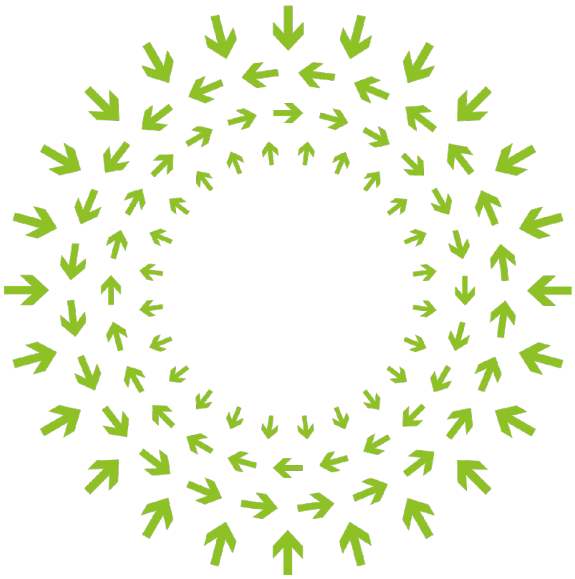


OPEN

Compute Project

Optics Reliability Workstream

Enhanced Telemetry Optics Product Software Specification

V0.9

Authors:
Credo: Chris Collins

Version History

Date	Version #	Author	Description
12 SEP 2025	0.8.0	Chris Collins	Initial Release
01 DEC 2025	0.8.1	Chris Collins	Updated format based on steering committee feedback
10 DEC 2025	0.9	Chris Collins	Changed to revision 0.9

Table of Contents

Version History **2**
Open Web Foundation (OWF) CLA **7**
Acknowledgements **8**
1 Compliance with OCP Tenets **9**
1.1 Openness 9
1.2 Efficiency 9
1.3 Impact 9
1.4 Scale 9
1.5 Sustainability 9
2 Scope **10**
3 Overview **11**
4 High Level Requirements Transceiver Requirements **12**
4.1 Responding to external queries 12
4.2 Module Status and Telemetry Metrics 12
4.3 Module Telemetry Data 13
4.4 Event Monitor and Logging 13
4.5 Other Mandatory Transceiver MCU Features 14
5 Transceiver CDB Implementation **15**
5.1 CDB Control Page Structure (Upper page 0x9F) 15
5.2 Extended Payload (EPL) (banked pages A0h - AFh, up to 2Kbytes) 15
5.3 CDB Command Status Registers (Lower Memory Map Bytes 37–38) 16
6 Capability Advertisement, Flag Reporting and Counter Mask **16**
6.1 Remote Capability Advertisement 16
6.2 Telemetry Flag Summary Reporting 17
6.3 Custom Mask for Link Health Score Counters 17
7 CDB Telemetry Info **18**
7.1 CDB Telemetry Info (CMD: 8000h) 18
7.1.1 Command Message Header (bytes 128 – 138) *18*
7.1.2 Reply Message Header (bytes 139 – 255) *19*
7.1.3 CDB Command Status (Page 0, byte 8 and bytes 37 – 38) *23*

7.2 Rating Score Formula (Link Health Score) 25
7.3 CDB Reset Telemetry Counter (CMD: 8001h) 27
7.3.1 Command Message Header (bytes 128 – 138) *27*
7.3.2 Reply Message Header *28*
7.3.3 CDB Command Status (Page 0, byte 8 and bytes 37 – 38) *28*
7.4 CDB Telemetry Counter Mask Control (CMD: 8002h) 29
7.4.1 Command Message Header (bytes 128 – 138) *30*
7.4.2 Reply Message Header *30*
7.4.3 CDB Command Status (Page 0, byte 8 and bytes 37 – 38) *31*
8 CDB Event Log Management 32
8.1 Get Event Log (CMD: 8010h, option = 0) 32
8.1.1 Event Log Entry Structure *33*
8.1.2 Event Log IDs and Types *33*
8.1.3 Event Log Threshold *34*
8.1.4 Command Message Header (bytes 128 – 142) *34*
8.1.5 Reply Message Header (bytes 136 – 244) *35*
8.1.6 CDB Command Status (Page 0, byte 8 and bytes 37 – 38) *38*
8.1.7 Event log parsing example *40*
8.2 Reset Event Log (CMD: 8010h, option = 1) 40
8.2.1 Command Message Header (bytes 128 – 141) *40*
8.3 Synchronize Event Log Time (CMD: 8011h) 41
8.3.1 Command Message Header (bytes 128 – 141) *41*
8.3.2 CDB Command Status (Page 0, byte 8 and bytes 37 – 38) *42*
9 CDB Remote CMIS Register Access 43
9.1 CDB Remote CMIS Byte Access (CMD: 8020h) 43
9.1.1 Command Message Header *44*
9.1.2 Reply Message Header *45*
9.1.3 CDB Command Status (Page 0, byte 8 and bytes 37 – 38) *45*
9.2 CDB Remote CMIS Page Access (CMD: 8021h) 47
9.2.1 Command Message Header *47*
9.2.2 Reply Message Header *48*
9.2.3 CDB Command Status (Page 0, byte 8 and bytes 37 – 38) *48*
10 CDB Threshold Control (CMD: 8030h) 50

11 CDB Firmware Management Features 50
11.1 CDB Firmware Management Features (Local) 50
11.2 CDB Firmware Management Features (Remote) 51
11.3 CDB Firmware Management Features (CMD: 8041h to F041h) 52
11.3.1 Reply Message Header 52
11.3.2 CDB Command Status (Page 0, byte 8 and bytes 37 – 38) 54
12 CDB Firmware Management 56
12.1 CDB Firmware Upgrade (Local) 56
12.1.1 Get Firmware information (CMD: 0100h) 57
12.1.2 Start Firmware Download (CMD: 0101h) 57
12.1.3 Abort Firmware Download (CMD: 0102h) 57
12.1.4 Firmware Data Transfer via LPL/EPL (CMD: 0103h/0104h) 58
12.1.5 Complete Firmware Download (CMD: 0107h) 58
12.1.6 Run Firmware Image (CMD: 0109h) 58
12.1.7 Commit Firmware Image (CMD: 010Ah) 58
12.2 CDB Firmware Upgrade (Remote) 58
12.2.1 Get Firmware information (CMD: 8100h to F100h) 60
12.2.2 Reply Message Header 61
12.2.3 CDB Command Status (Page 0, byte 8 and bytes 37 – 38) 63
12.3 Start Remote Firmware Download (CMD: 8101h to F101h) 64
12.3.1 Reply Message Header 65
12.3.2 CDB Command Status (Page 0, byte 8 and bytes 37 – 38) 65
12.4 Abort Remote Firmware Download (CMD: 8102h to F102h) 67
12.4.1 Reply Message Header 68
12.4.2 CDB Command Status (Page 0, byte 8 and bytes 37 – 38) 68
12.5 Remote Firmware Data Transfer (CMD: 8104h to F104h) 69
12.5.1 Reply Message Header 70
12.5.2 CDB Command Status (Page 0, byte 8 and bytes 37 – 38) 70
12.6 Complete Remote Firmware Download (CMD 8107h to F107h) 72
12.6.1 Reply Message Header 72
12.6.2 CDB Command Status (Page 0, byte 8 and bytes 37 – 38) 73
12.7 Run Remote Firmware Image (CMD: 8109h to F109h) 74
12.7.1 Reply Message Header 75

12.7.2 CDB Command Status (Page 0, byte 8 and bytes 37 – 38) 75
12.8 Commit Remote Firmware Image (CMD: 810Ah to F10Ah)..... 77
12.8.1 Reply Message Header..... 77
12.8.2 CDB Command Status (Page 0, byte 8 and bytes 37 – 38) 77
13 Host Software 80
13.1 Telemetry SDK 80
13.2 Transceiver Communications..... 80
13.3 Command Line APIs..... 80
14 Requirements and Repository..... 81
14.1 Repository Location 81
Appendix A - Checklist for Steering Committee (SC) approval of this Specification (to be completed by contributor(s) of this Spec) 82

THE UPDATED DEFAULT CONTRIBUTOR LICENSE AGREEMENT (CLA) IS **OWFa 0.9**. PLEASE VERIFY THE CORRECT CLA/FSA IS USED AND EXECUTED FOR THIS CONTRIBUTION.

Open Web Foundation (OWF) CLA

Contributions to this Specification are made under the terms and conditions set forth in **Modified Open Web Foundation Agreement 0.9 (OWFa 0.9)**. (As of October 16, 2024) ("Contribution License") by:

Credo Semiconductor

Usage of this Specification is governed by the terms and conditions set forth in **Modified OWFa 0.9 Final Specification Agreement (FSA)** (As of October 16, 2024) **("Specification License").**

You can review the applicable Specification License(s) referenced above by the contributors to this Specification on the OCP website at https://www.opencompute.org/contributions/templates-agreements.

For actual executed copies of either agreement, please contact OCP directly.

Notes:

> The above license does not apply to the Appendix or Appendices. The information in the Appendix or Appendices is for reference only and non-normative in nature.

NOTWITHSTANDING THE FOREGOING LICENSES, THIS SPECIFICATION IS PROVIDED BY OCP "AS IS" AND OCP EXPRESSLY DISCLAIMS ANY WARRANTIES (EXPRESS, IMPLIED, OR OTHERWISE), INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR A PARTICULAR PURPOSE, OR TITLE, RELATED TO THE SPECIFICATION. NOTICE IS HEREBY GIVEN, THAT OTHER RIGHTS NOT GRANTED AS SET FORTH ABOVE, INCLUDING WITHOUT LIMITATION, RIGHTS OF THIRD PARTIES WHO DID NOT EXECUTE THE ABOVE LICENSES, MAY BE IMPLICATED BY THE IMPLEMENTATION OF OR COMPLIANCE WITH THIS SPECIFICATION. OCP IS NOT RESPONSIBLE FOR IDENTIFYING RIGHTS FOR WHICH A LICENSE MAY BE REQUIRED IN ORDER TO IMPLEMENT THIS SPECIFICATION. THE ENTIRE RISK AS TO IMPLEMENTING OR OTHERWISE USING THE SPECIFICATION IS ASSUMED BY YOU. IN NO EVENT WILL OCP BE LIABLE TO YOU FOR ANY MONETARY DAMAGES WITH RESPECT TO ANY CLAIMS RELATED TO, OR ARISING OUT OF YOUR USE OF THIS

SPECIFICATION, INCLUDING BUT NOT LIMITED TO ANY LIABILITY FOR LOST PROFITS OR ANY CONSEQUENTIAL, INCIDENTAL, INDIRECT, SPECIAL OR PUNITIVE DAMAGES OF ANY CHARACTER FROM ANY CAUSES OF ACTION OF ANY KIND WITH RESPECT TO THIS SPECIFICATION, WHETHER BASED ON BREACH OF CONTRACT, TORT (INCLUDING NEGLIGENCE), OR OTHERWISE, AND EVEN IF OCP HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This document references and utilizes information from **OIF-CMIS-05.3** and as such is subject to OIF terms.

Acknowledgements

The Contributors of this Specification would like to acknowledge the following for their feedback:

Stephen Manley – Oracle
Kannan Raj – Oracle
Charles Park - Oracle
Evan Lin – Credo
JJ Wang – Credo
Mike Kao – Credo

1 Compliance with OCP Tenets

A full explanation of the OCP core tenets can be seen here.

1.1 Openness

By contributing the Enhanced Telemetry Optics Software Specification to the Open Compute Foundation, it allows all members to enjoy the benefits of increased reliability and fewer link flaps in large scale AI networks.

1.2 Efficiency

The goal of contributing the Enhanced Telemetry Optics Software Specification is to drive commonality in various optical transceiver implementations. This will enable more efficient qualifications and faster optical deployment by directly reducing the OPEX required by eliminating the need to rewrite host software for each different transceiver. Furthermore, by increasing the reliability and uptime of large-scale GPU clusters, operational efficiencies are enhanced with fewer system restarts, faster LLM training and less manual system debug.

1.3 Impact

Optical transceivers are a critical component in deploying AI fabrics. By helping to enhance their reliability, system operators benefit from fewer network failures and improved operational performance. When bare-metal server access is provided to end users, network operators lose visibility into optical endpoints leaving them blind in the event of a link problem. With remote management capability, compliant optics provide visibility right to the transceiver in the bare metal server. As GPU clusters grow to hundreds of thousands, or even millions of bare-metal compute instances, providing visibility into server link quality across an entire fabric becomes a fundamental necessity.

1.4 Scale

Optical transceivers are being deployed at unprecedented rates in RDMA fabrics for AI. The shipment of 400G, 800G and 1.6T optical transceivers is expected to approach 50 million units per year in 2026. This software specification specifically targets these and future, higher rate, optical transceivers.

1.5 Sustainability

This contribution is specifically focused on making large-scale AI deployments more efficient by decreasing downtime due to optical networking issues. By making the Enhanced Telemetry Optics Software Specification open source, the goal is to ensure that the entire ecosystem of transceiver vendors and network switch vendors can meet the industry demand for equipment.

2 Scope

The purpose of this document is to define a software stack that provides a common means of collecting and extracting specific telemetry information from optical transceivers deployed in AI fabrics. The requirements herein are applicable to 400G, 800G and 1.6T compliant transceivers and are intended to be independent of the module PMD. There are three (3) components to a compliant software system.

1. Transceiver firmware and telemetry metrics
2. Host software (typically residing on a switch)
3. Telemetry server software used to compile, summarize and report overall network health

The focus of this document is on the transceiver implementation and providing a common interface to the host. The Telemetry server software implementation is user dependent and NOT within the scope of this document.

This specification uses many of the transceiver functions defined in OIF-CMIS-05.x and adds specific definition and modifications above and beyond CMIS to achieve the stated goals.

.

3 Overview

Compliant transceivers integrate six key functions to facilitate enhanced link visibility and management.

1. Mission mode link monitoring and specific statistics collection on host and line facing receivers
2. A common link health reporting metric along with real time tracking and logging of link telemetry data
3. Transparent, mission mode, in-band communication for remote identification and data exchange between the transceivers at either end of a link.
4. Programmable event notification and non-volatile event logging in transceiver memory
5. Capability for hitless transceiver firmware update on both local and remote transceivers
6. Switch software to enable harvesting of transceiver telemetry, storage and preparation for streaming to a telemetry server

The combination of these six functions allows a network engineer to answer the following questions:

- Are my links up?
    - For links that are up:
        - Who is the link partner?
        - Are the transceivers at both ends reporting healthy links?
        - What is the overall health of the link?
        - Is service required?
        - If service is needed, which portions of the links are unhealthy?
    - If a link flaps, why did it flap?
        - Was it intentional?
        - What caused the flap?

A point-to-point optical link is generally made up of six (6) link segments. Compliant optical transceivers along with the host ports, enable independent monitoring of each link segment. In the event of an issue, this information allows for quick diagnosis of the degraded component and expedites resolution. In addition, in-band remote communication enables telemetry harvesting from each segment by the switch without requiring direct access to the transceiver at the server side of the link.

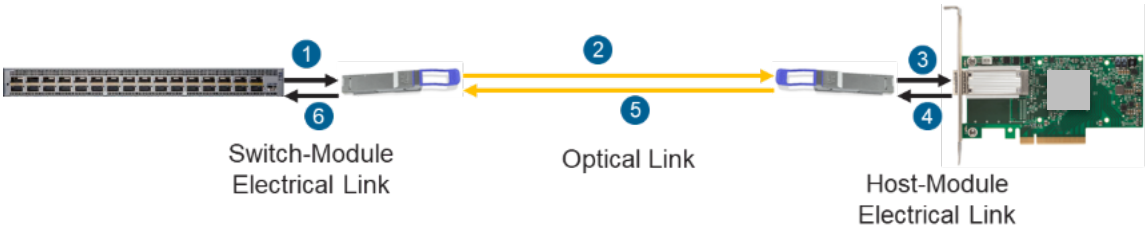


Figure 1: Network Link with two Optical Transceivers

In some instances, breakout links are used where a single switch port may connect to two or more host ports. In this case, each of the link segments will be monitored independently and the module in the switch shall harvest telemetry data from each of the remote link partners using independent messaging interfaces for each link partner.

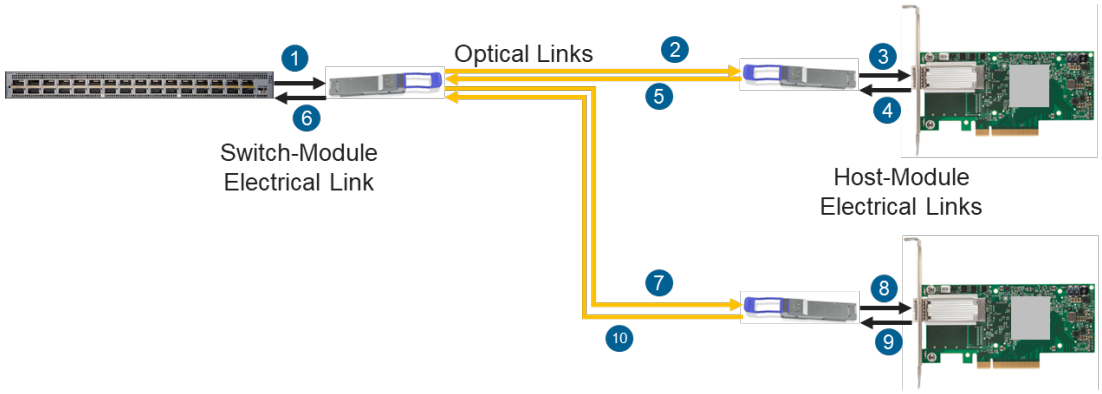


Figure 2: Breakout connection example with two hosts

4 High Level Requirements Transceiver Requirements

This section provides high level details on the transceiver requirements for compliant functionality.

The MCU inside the compliant transceiver must provide an interface for the host to query transceiver status information for both the local transceiver and remote link partner(s) at the far end of the fiber.

4.1 Responding to external queries

The transceiver MCU must respond to external requests generated locally via the module I2C interface or generated remotely via in-band messaging interface from the far end link partner.

4.2 Module Status and Telemetry Metrics

The following is a list of module status and telemetry metrics that must be made available by the transceiver to the host system. The transceiver must be able to gather the following information **locally or from any of its remote link partners** over the in-band messaging interface. NOTE: a transceiver may be connected to more than one link partner in the case of a breakout use case. (i.e. 800G to 2 x 400G).

a) All module CMIS Bank 0 Pages 00h – 0Fh
b) CMIS Configuration and Status Pages 0..N[10h-1Fh]
c) CMIS VDM Pages 0..N[20h-2Fh]
d) CMIS OIF Restricted Pages 0..N[30h-3Fh]
e) CMIS OIF Restricted Pages 0[40h-4Fh]
f) CMIS Reserved Pages 0[50h-9Eh]

g) CMIS CDB Command Pages 0..1[9Fh]
h) CMIS CDB Payload Pages 0..1[A0h-AFh]
i) CMIS Custom Vendor Page 0 [B0h-FFh]

4.3 Module Telemetry Data

Many of the link parameters defined in this section are also defined in the CMIS 5.x standard under the VDM section. The data collected for each of the measurements listed below shall be stored in the transceiver non-volatile memory (memory map provided in this document). The transceiver MCU shall update each of these parameters periodically, with a programmable rate between 1 second and 10 seconds.

a) Signal to Noise (SNR) Measurement on the optical (line) link for each lane
b) Signal to Noise (SNR) Measurement on the electrical (host) link for each lane
c) Eye height measurement for each lane (host and line side)
d) PAM4 level transition parameter (LTP) for each lane (host and line side)
e) MPI estimation for each optical lane
f) Pre-FEC BER measurement
g) FEC Bin histogram (accumulated errors per FEC frame)
h) Count of un-correctable codewords
i) Count of relinks (link drop and reacquisition)
j) Loss of Signal (LOS) and Loss of Link (LOL)
k) Channel quality metric (built using a combination of the metrics above)
l) “Uptime”- elapsed time since last new link up on the host and line interfaces
m) Module temperature
n) Laser temperature
o) DDMI Alarm/Warning Flag

4.4 Event Monitor and Logging

Major events shall be logged in a non-volatile circular buffer inside of each transceiver so that they can be retrieved for historical debug even after the transceiver has been removed and re-inserted in a different host system. On power up, it is essential that the transceiver MCU does NOT overwrite data previously stored in the non-volatile circular buffer. This ensures the user can retrieve data from recent failure events even if the module is extracted and plugged in elsewhere.

Major events shall be logged with a corresponding timestamp. Major events include:

- Power-up (with no time stamp)
- Write Epoch time (from local or remote host)
- The events below should be logged separately for host side and optical side of the transceiver with a time stamp.

- Link-up
    - Loss of Signal (LOS) indicating which receiver
    - Loss of Link (LOL) indicating which receiver
    - MPI exceeds a programmable threshold includes the MPI value
        - The threshold should be programmable from local or remote side
        - This parameter only applies to the optical interface
    - SNR falls below a programmable threshold and the actual SNR
        - The threshold should be programmable from local or remote side
    - Pre-FEC BER exceeds a programmable threshold (with calculated BER)
        - This threshold has a default of 1e-8 and is programmable from the local or remote side
    - FEC histogram exceeds a programmable bin threshold (with maximum observed FEC bin)
        - This threshold has a default (bin 6) and is programmable from the local or remote side
    - Uncorrectable FEC codeword is reported (report the number of uncorrectable since last read)
    - Module temperature exceeds a programmable threshold and the measured temperature
    - Received optical power exceeds a programmable threshold and the measured Rx power in dBm
- Link-down
- Firmware image overwrite (A/B)
- Firmware image make active (A/B)
- Soft reboot request

As Link-up / down events and Pre-FEC BER / histogram events can happen repeatedly, there is a risk of over-running the circular buffer with events and making debug less useful. Hysteresis should be added to prevent duplicate events within a programmable time period (default 10 seconds).

The event monitor can be accessed locally from the I2C port, remotely via the in-band messaging and locally or remotely using the switch side SDK.

4.5 Other Mandatory Transceiver MCU Features

a) The transceiver shall record Epoch time to the local transceiver at each power up so that events can be time-stamped with a baseline reference
b) The module shall be capable of linkup with both another book-ended compliant transceiver (including in-band messaging) as well as 3rd party optics (without enhanced telemetry features).
c) If the module links to another compliant transceiver, it shall propagate Epoch time to the far side transceiver(s) if they have not already had Epoch time locally written by their hosts
d) Transceivers shall support hemispherical firmware update mode – with ability to upgrade the firmware in a hitless manner.

e) The transceiver shall support remote firmware update for the far end transceiver over the in-band communication interface

5 Transceiver CDB Implementation

Each of the telemetry functions detailed in Section 5 is implemented in such a way as to ensure efficient telemetry harvesting by the host and over the in-band communication interface. Command Data Block (CDB) is an optional CMIS feature for command-reply message exchange between host and module. A transceiver compliant with this specification shall use the CDB feature exclusively for telemetry collection and transfer to the host system.

A typical CDB command sequence consists of four main steps:

1. Prepare the command message by filling in the required header fields and payload.
2. Trigger the command by writing the message to the appropriate location.
3. Poll the CDB status registers to monitor command execution progress.
4. Retrieve the reply message from the reply header and payload area once the command has completed.

5.1 CDB Control Page Structure (Upper page 0x9F)

Page 9Fh is used for CDB messaging between host and module. Reference CMIS 5.3 section 8.23 Banked Page 9Fh (CDB Message) for detailed information on the register field definitions. The following figures in the CMIS document are particularly relevant.

Table 8-177 – Page 9Fh Overview
Table 8.178 – CDB Command Message Header (Bytes 128 – 133)
Table 8.179 – CDB Reply Message Header (Bytes 134 – 135)
Table 8.180 – CDB Message Body (Bytes 136 – 255) – Local Payload (LPL)
Table 8.181 – CDB Extended Payload (EPL) - (page A0h ~ AFh, up-to 2Kbytes)

5.2 Extended Payload (EPL) (banked pages A0h - AFh, up to 2Kbytes)

This area is used for large payloads (up to 2048 bytes) and is addressed in additional pages (A0h–AFh) when the LPL is insufficient.

- For larger payloads (up to 2048 bytes), additional pages (A0h–AFh) are required
- EPLLength specifies the number of bytes in EPL.

5.3 CDB Command Status Registers (Lower Memory Map Bytes 37–38)

These registers shall be updated by the module and shall be read by the host after each command to determine execution status and completion.

After triggering a CDB command, the user shall read and check Page 00h bytes 37 and 38 to determine the command's completion and status. While processing a CDB command, the module shall update the command execution status in CdbStatus 00h:38 or 00h:37 and signals completion in the CDB CMD Completion Flags 00h:8.7 or 00h:8.6.

Note:

- Byte 37: bank 0 CDB status
- Byte 38: bank 1 CDB status

6 Capability Advertisement, Flag Reporting and Counter Mask

This section defines the module's remote capability advertisement mechanism, custom flag summary reporting system for port monitoring, and custom counter mask configuration for the transceiver health score calculation

6.1 Remote Capability Advertisement

The remote capability field in byte 36 is used to report whether the module supports remote applications (via in-band messaging interface) and to specify the total number of remote ports to use. Note that for each remote port, an associated local port will also be defined. For example, if a single 800G transceiver is assigned two 400G remote ports, then the 800G transceiver will be virtually split into 2 x 400G ports as well.

Page	Byte	Bit	Name	Field Description
0	36	7-4	Remote Ports	Total number of remote ports (number of link partners) that can be accessed by remote communication
		3-1	-	Reserved
		0	Remote capability	Remote application support indicator for in-band messaging 0b: not supported 1b: supported

6.2 Telemetry Flag Summary Reporting

A custom flag in byte 13 shall be used as a summary of remote and local port flags. The transceiver shall set any corresponding bit if a flag is present on the indicated module and port. The recommended query procedure is to follow the standard module flag handling process. First, read the software interrupt (byte 3, bit 0); then, read byte 13 to determine which ports triggered the flag reporting; finally, parse bytes 162–164 of the corresponding reply message of telemetry data to identify which flags were set.

Page	Byte	Bit	Name	Field Description
0	13	7	Flag Summary of Remote Ports	1b: at least one Flag is set on remote port 3
		6		1b: at least one Flag is set on remote port 2
		5		1b: at least one Flag is set on remote port 1
		4		1b: at least one Flag is set on remote port 0
		3	Flag Summary of Local Ports	1b: at least one Flag is set on local port 3
		2		1b: at least one Flag is set on local port 2
		1		1b: at least one Flag is set on local port 1
		0		1b: at least one Flag is set on local port 0

6.3 Custom Mask for Link Health Score Counters

The custom mask in byte 64 determines which counter types from the Link Health Score defined in Section 8.1.4 are excluded from rating score calculation. Mask bit value 1 indicates excluded, while value 0 indicates included.

Page	Byte	Bit	Name	Field Description
0	64	7	-	Reserved
		6		
		5		
		4		
		3		
		2		
		1	FecCounterMask	Mask bit for FEC Counter (excluded by default)
		0	LinkDownCounterMask	Mask bit for Link Down Counter (excluded by default)

7 CDB Telemetry Info

A transceiver aggregates all key link health information in a single CDB telemetry info command page for efficient data transfer. This includes:

- Port health rating score
- Module serial number
- Link down counter and link up time
- Module level and optics flags
- Signal-to-noise ratio (SNR) and PAM Level Transition Parameter (LTP)
- Eye height statistics (lower, middle, upper) in mV
- Module temperature, voltage and optical receiver power
- Pre-FEC BER and Forward Error Correction (FEC) bin counts

7.1 CDB Telemetry Info (CMD: 8000h)

The telemetry info command (CMDID: 8000h) is used to request and retrieve detailed telemetry and diagnostic information from the module. This command allows the host to access a wide range of real-time status and performance metrics for monitoring and troubleshooting purposes. Note this command is a custom CDB command ID that shall be supported by transceivers complying with this specification.

7.1.1 Command Message Header (bytes 128 – 138)

To trigger the 0x8000 telemetry command, the host shall write the CDB command message as follows:

- Fill the command argument to config the target, side and port
- Set LPL length and CDB check code
- Set CMDID to 0x8000 to initiate the telemetry command.
- Attributes are in **big-endian** format

CDB Telemetry Command Message			
Page	**Byte**	**Attribute**	**Description**
9Fh	128	CMDID	0x8000
	129		
	130	EPL length	N/A
	131		
	132	LPL length	VAR
	133	Cdb chk code	VAR
	134	RPL length	N/A

CDB Telemetry Command Message			
Page	**Byte**	**Attribute**	**Description**
	135	RPL chk code	N/A
	136	Target	selection for local or remote end 0: local end 1: remote end
	137	Side	selection for cable side 0: host side 1: media side
	138	Port	port number, e.g. for a 2x200G 0: 1st 200G 1: 2nd 200G

7.1.2 Reply Message Header (bytes 139 – 255)

Once the telemetry command has been successfully processed by the module, the user application can retrieve the information listed below

CDB Telemetry Reply Message			
Page	**Byte**	**Attribute**	**Description**
9Fh	134	RPL length	VAR
	135	RPL chk code	VAR
	136	Module Status	bit 0: link status. 1 = Target is online and accessible. 0 = Target is offline and not accessible. bit 1 - 7: Reserved
	137	Lowest Rating Category Mask (MSB)	16-bit bitmask (MSB). If a bit is set, that category is one of the worst contributors. Multiple bits may be set if multiple categories share the lowest rating score bit7- 0: Reserved
	138	Lowest Rating Category Mask (LSB)	16-bit bitmask (LSB). If a bit is set, that category is one of the worst contributors. Multiple bits may be set if multiple categories share the lowest rating score. bit7: Reserved bit6: All flags bit5: Eye Height bit4: SNR bit3: MPI bit2: Pre-FEC BER bit1: FEC Bin Counter bit0: Link Down Counter

CDB Telemetry Reply Message			
Page	**Byte**	**Attribute**	**Description**
	139	Rating Score	Overall rating score for the port health status, serving as a quick reference indicator. - Critical: Score <= 40 - Good: Score = 41 - 79 - Excellent: Score >= 80
	140	Serial Number	Module serial number
	141		
	142		
	143		
	144		
	145		
	146		
	147		
	148		
	149		
	150		
	151		
	152		
	153		
	154		
	155		
	156	Link Down Counter	Serdes link down counter (This value indicates a Layer 1 PHY down condition, such as the absence of received optical signal)
	157		
	158	Link Up Time	Number of seconds that have elapsed since last link up
	159		
	160		
	161		
	162	Module Flag	These registers capture and latch warning and alarm events for module voltage, temperature. Reading the register will clear all bits, resetting the status until a new event occurs. bit7: Vcc monitor low warning flag bit6: Vcc monitor high warning flag bit5: Vcc monitor low alarm flag bit4: Vcc monitor high alarm flag bit3: Temp monitor low warning flag

CDB Telemetry Reply Message			
Page	**Byte**	**Attribute**	**Description**
			bit2: Temp monitor high warning flag bit1: Temp monitor low alarm flag bit0: Temp monitor high alarm flag
	163	Optical Power Flag	These registers capture and latch warning and alarm events for optical power. Reading the register will clear all bits, resetting the status until a new event occurs. bit7: Optical rx power low warning flag bit6: Optical rx power high warning flag bit5: Optical rx power low alarm flag bit4: Optical rx power high alarm flag bit3: Optical tx power low warning flag bit2: Optical tx power high warning flag bit1: Optical tx power low alarm flag bit0: Optical tx power high alarm flag
	164	Optical Laser Bias Flag	These registers capture and latch warning and alarm events for laser bias. Reading the register will clear all bits, resetting the status until a new event occurs. bit3: Optical tx laser bias low warning flag bit2: Optical tx laser bias high warning flag bit1: Optical tx laser bias low alarm flag bit0: Optical tx laser bias high alarm flag
	165	MPI Detection	Multipath interference value
	166	SNR	The worst lane's SNR value in units of 1/256 dB.
	167		
	168	LTP	The worst lane's LTP value in units of 1/256 dB.
	169		
	170	Eye Height (Lower)	Eye height (in mV) of worst lane
	171		
	172	Eye Height (Middle)	
	173		
	174	Eye Height (Upper)	
	175		
	176	Module Temperature	Module temperature monitor in 1/256 degree celsius
	177		
	178	Module Voltage	Supply voltage monitors in 100 uV

CDB Telemetry Reply Message			
Page	**Byte**	**Attribute**	**Description**
	179		
	180	Optical Rx Power (Min of all channels)	Optical Rx power in 0.1 uW increments, yielding a total measurement range of 0 to 6.5535 mW (-40.0 to +8.2 dBm for non-zero values)
	181		
	182	Optical Rx Power (Max of all channels)	
	183		
	184	PreFEC BER	Bit7-3: Scaled Exponent (s) Bit2-0: Mantissa (m), most significant bits 10-8
	185		Mantissa (m), least significant bits 7-0
	186	Post-Fec Errors	Number of uncorrected errors detected at the receiver.
	187		
	188		
	189		
	190	FEC bin 0	Number of FEC frames with zero post-FEC errors
	191		
	192		
	193		
	194		
	195		
	196		
	197		
	198	FEC bin 1	Number of FEC frames with 1 symbol error
	199		
	200		
	201		
	202	FEC bin 2	Number of FEC frames with 2 symbol errors
	203		
	204		
	205		
	206	FEC bin 3	Number of FEC frames with 3 symbol errors
	207		
	208		
	209		
	210	FEC bin 4	Number of FEC frames with 4 symbol errors
	211		
	212		
	213		
	214	FEC bin 5	Number of FEC frames with 5 symbol errors

CDB Telemetry Reply Message			
Page	**Byte**	**Attribute**	**Description**
	215		
	216		
	217		
	218	FEC bin 6	Number of FEC frames with 6 symbol errors
	219		
	220		
	221		
	222	FEC bin 7	Number of FEC frames with 7 symbol errors
	223		
	224		
	225		
	226	FEC bin 8	Number of FEC frames with 8 symbol errors
	227		
	228	FEC bin 9	Number of FEC frames with 9 symbol errors
	229		
	230	FEC bin 10	Number of FEC frames with 10 symbol errors
	231		
	232	FEC bin 11	Number of FEC frames with 11 symbol errors
	233		
	234	FEC bin 12	Number of FEC frames with 12 symbol errors
	235		
	236	FEC bin 13	Number of FEC frames with 13 symbol errors
	237		
	238	FEC bin 14	Number of FEC frames with 14 symbol errors
	239		
	240	FEC bin 15	Number of FEC frames with 15 symbol errors
	241		
	242	Custom Counter Mask	Sync the custom counter mask value as defined in section 7.3.
	243-255	Reserved	

7.1.3 CDB Command Status (Page 0, byte 8 and bytes 37 – 38)

These registers are updated by the module and must be read by the host after each command to determine execution status and completion

CDB Command Status				
Page	**Byte**	**Bit**	**Attribute**	**Description**
0h	8	6	CdbCmdCompleteFlag1	Latched Flag to indicate completion of a CDB command for CDB instance 1
		7	CdbCmdCompleteFlag2	Latched Flag to indicate completion of a CDB command for CDB instance 2
	37	7 - 0	CdbStatus1	Status of the most recent CDB command in CDB instance 1
	38	7 - 0	CdbStatus2	Status of the most recent CDB command in CDB instance 2

<table>
<tr><th colspan="3">CdbStatus</th></tr>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>5 – 0</td><td>CdbCommandResult</td><td>The CdbCommandResult field provides more detailed classification for each of the three coarse query results that are encoded by the pair of bit 7 (CdbIsBusy) and bit 6 (CdbHasFailed)

<table>
<tr><td>Coarse Status</td><td>CdbIsBusy</td><td>CdbHasFailed</td></tr>
<tr><td>IN PROGRESS</td><td>1</td><td>X (don’t care)</td></tr>
<tr><td>SUCCESS</td><td>0</td><td>0</td></tr>
<tr><td>FAILED</td><td>0</td><td>1</td></tr>
</table>

The interpretation of CdbCommandResult therefore depends on the coarse status as follows:
IN PROGRESS
00h=Reserved
01h=Command is captured but not processed
02h=Command checking is in progress
03h=Command execution is in progress
04h-2Fh=Reserved
30h-3Fh=Custom
SUCCESS
00h=Reserved
01h=Command completed successfully
02h=Reserved
03h=Previous CMD was ABORTED by CMD Abort
04h-1Fh=Reserved
20h-2Fh=Reserved
30h-3Fh=Custom
FAILED
00h=Reserved</td></tr>
</table>

CdbStatus		
Bit	**Attribute**	**Description**
		01h=CMDID unknown 02h=Parameter range error or parameter not supported 03h=Previous CMD was not properly ABORTED (by CMD Abort) 04h=Command checking time out 05h=CdbChkCode Error 06h=Password related error (command specific meaning) 07h=Command not compatible with operating status 08h-0Fh=Reserved for STS command checking error 10h-1Fh=Reserved 20h-2Fh=For individual STS command or task error 30h-3Fh=Custom
6	CdbHasFailed	Bool: CdbHasFailed bit indicates if there was a failure, after the module has completed execution of the last CDB command 0b: Last triggered CDB command completed successfully 1b: Last triggered CDB command failed
7	CdbIsBusy	Bool: CdbIsBusy status bit indicates whether the module is still busy, or idle and ready to accept a new CDB command 0b: Module idle, host can write 1b: Module busy, host needs to wait

7.2 Rating Score Formula (Link Health Score)

The rating score provides a quantitative assessment of port health based on key optical and electrical parameters. It is evaluated against defined thresholds. The condition of each category is assessed first, then the minimum score from all categories is selected as the final score. Averaging of the fields is not used as a single bad field (such as a high pre-FEC BER) could be accidentally masked if it were averaged with many other acceptable categories.

Category	Field Name	Condition	Score
1	Link Down Counter*	0	80
		>=1	40
2	Module flag	no flag	80
		warning flags	60
		alarm flags	40
3	Optical Power Flag	no flag	80
		warning flags	60
		alarm flags	40
4	Optical Laser Bias Flag	no flag	80

Category	Field Name	Condition	Score
		warning flags	60
		alarm flags	40
5	SNR	>18 dB	80
		14-18 dB	60
		<14 dB	40
6	LTP	>45 dB	80
		35-45 dB	60
		<35 dB	40
7	Eye Height	> 50mV	80
		30-50mV	60
		<30mV	40
8	Module Temperature	>75 C	40
		-5 ~ 75C	60
		<-5C	40
9	Module Voltage	<2.8V	40
		2.8~3.13	60
		3.13~3.46V	80
		3.46~3.6	60
		>3.6V	40
10	Rx Optical Power	> -3 dBm	80
		-6 ~ -3dBm	60
		< -6 dBm	40
11	Pre-FEC BER	<1e-8	80
		1e-5~1e-8	60
		>1e-5	40
12	Post-FEC Frame Error*	0	80
		>1	40
13	FEC – Maximum Corrected Bin	FEC bin < 4	80
		FEC bin 4 ~ 7	60

Category	Field Name	Condition	Score
		FEC bin > 7	40

* fields that are excluded by default from the Link Health Score (see Section 7.3)

7.3 CDB Reset Telemetry Counter (CMD: 8001h)

The Reset Telemetry Counter command (CMDID: 8001h) is used to reset the accumulated Forward Error Correction (FEC) bin counters and the module link-down counter within the telemetry data.

7.3.1 Command Message Header (bytes 128 – 138)

To trigger the 0x8001 Reset Telemetry Counter command, the host must fill in the command message as follows:

- Fill the command argument to config the target, side and port
- Set LPL length and CDB check code
- Set CMDID to 0x8001 to initiate the Reset Telemetry Counter command

CDB Reset FEC Bin Counter Message			
Page	**Byte**	**Attribute**	**Description**
9Fh	128	CMDID	0x8001
	129		
	130	EPL length	N/A
	131		
	132	LPL length	VAR
	133	Cdb chk code	VAR
	134	RPL length	N/A
	135	RPL chk code	N/A
	136	Target	selection for local or remote end 0: local end 1: remote end
	137	Side	selection for cable side 0: host side 1: media side
	138	Port	port number, e.g. for a 2x200G 0: 1st 200G 1: 2nd 200G

7.3.2 Reply Message Header

CDB Reset FEC Bin Counter Command (Reply)	
Reply	**Description**
RPL length	VAR
RPL chk code	VAR

7.3.3 CDB Command Status (Page 0, byte 8 and bytes 37 – 38)

These registers are updated by the module and must be read by the host after each command to determine execution status and completion.

CDB Command Status				
Page	**Byte**	**Bit**	**Attribute**	**Description**
0h	8	6	CdbCmdCompleteFlag1	Latched Flag to indicate completion of a CDB command for CDB instance 1
		7	CdbCmdCompleteFlag2	Latched Flag to indicate completion of a CDB command for CDB instance 2
	37	7 - 0	CdbStatus1	Status of the most recent CDB command in CDB instance 1
	38	7 - 0	CdbStatus2	Status of the most recent CDB command in CDB instance 2

<table>
<tr><th colspan="3">CdbStatus</th></tr>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>5 – 0</td><td>CdbCommandResult</td><td>The CdbCommandResult field provides more detailed classification for each of the three coarse query results that are encoded by the pair of bit 7 (CdbIsBusy) and bit 6 (CdbHasFailed)

<table>
<tr><td>Coarse Status</td><td>CdbIsBusy</td><td>CdbHasFailed</td></tr>
<tr><td>IN PROGRESS</td><td>1</td><td>X (don’t care)</td></tr>
<tr><td>SUCCESS</td><td>0</td><td>0</td></tr>
<tr><td>FAILED</td><td>0</td><td>1</td></tr>
</table>

The interpretation of CdbCommandResult therefore depends on the coarse status as follows:</td></tr>
</table>

CdbStatus		
Bit	**Attribute**	**Description**
		IN PROGRESS 00h=Reserved 01h=Command is captured but not processed 02h=Command checking is in progress 03h=Command execution is in progress 04h-2Fh=Reserved 30h-3Fh=Custom **SUCCESS** 00h=Reserved 01h=Command completed successfully 02h=Reserved 03h=Previous CMD was ABORTED by CMD Abort 04h-1Fh=Reserved 20h-2Fh=Reserved 30h-3Fh=Custom **FAILED** 00h=Reserved 01h=CMDID unknown 02h=Parameter range error or parameter not supported 03h=Previous CMD was not properly ABORTED (by CMD Abort) 04h=Command checking time out 05h=CdbChkCode Error 06h=Password related error (command specific meaning) 07h=Command not compatible with operating status 08h-0Fh=Reserved for STS command checking error 10h-1Fh=Reserved 20h-2Fh=For individual STS command or task error 30h-3Fh=Custom
6	CdbHasFailed	Bool: CdbHasFailed bit indicates if there was a failure, after the module has completed execution of the last CDB command 0b: Last triggered CDB command completed successfully 1b: Last triggered CDB command failed
7	CdbIsBusy	Bool: CdbIsBusy status bit indicates whether the module is still busy, or idle and ready to accept a new CDB command 0b: Module idle, host can write 1b: Module busy, host needs to wait

7.4 CDB Telemetry Counter Mask Control (CMD: 8002h)

The Telemetry Counter Mask Control command (CMDID: 8002h) is used to control which counter types from section 8.1.4 are excluded or included from rating score calculation.

7.4.1 Command Message Header (bytes 128 – 138)

To trigger the 0x8002 Telemetry Counter Mask Control command, the host must fill in the command message as follows:

- Fill the command argument to config the target, port, and counter mask value
- Set LPL length and CDB check code
- Set CMDID on 0x8002 to initiate the Reset Telemetry Counter Mask command.

CDB Reset Telemetry Counter Mask Message			
Page	**Byte**	**Attribute**	**Description**
9Fh	128	CMDID	0x8002
	129		
	130	EPL length	N/A
	131		
	132	LPL length	VAR
	133	Cdb chk code	VAR
	134	RPL length	N/A
	135	RPL chk code	N/A
	136	Target	selection for local or remote end 0: local end 1: remote end
	137	Reserved	reserved
	138	Port	port number, e.g. for a 2x200G 0: 1st 200G 1: 2nd 200G
	139	Counter Mask Value	counter mask value to set, (e.g. value 0x01 means only the link down counter is excluded from the rating score calculation). bit 0: LinkDownCounterMask bit 1: FecCounterMask bit 2 - 7: Reserved

7.4.2 Reply Message Header

CDB Reset Telemetry Counter Mask Reply Message			
Page	**Byte**	**Attribute**	**Description**
9Fh	134	RPL length	VAR
	135	RPL chk code	VAR

7.4.3 CDB Command Status (Page 0, byte 8 and bytes 37 – 38)

These registers are updated by the module and must be read by the host after each command to determine execution status and completion

CDB Command Status				
Page	**Byte**	**Bit**	**Attribute**	**Description**
0h	8	6	CdbCmdCompleteFlag1	Latched Flag to indicate completion of a CDB command for CDB instance 1
		7	CdbCmdCompleteFlag2	Latched Flag to indicate completion of a CDB command for CDB instance 2
	37	7 - 0	CdbStatus1	Status of the most recent CDB command in CDB instance 1
	38	7 - 0	CdbStatus2	Status of the most recent CDB command in CDB instance 2

<table>
<tr><th colspan="3">CdbStatus</th></tr>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>5 – 0</td><td>CdbCommandResult</td><td>The CdbCommandResult field provides more detailed classification for each of the three coarse query results that are encoded by the pair of bit 7 (CdbIsBusy) and bit 6 (CdbHasFailed)

<table>
<tr><td>Coarse Status</td><td>CdbIsBusy</td><td>CdbHasFailed</td></tr>
<tr><td>IN PROGRESS</td><td>1</td><td>X (don’t care)</td></tr>
<tr><td>SUCCESS</td><td>0</td><td>0</td></tr>
<tr><td>FAILED</td><td>0</td><td>1</td></tr>
</table>
The interpretation of CdbCommandResult therefore depends on the coarse status as follows:
IN PROGRESS
00h=Reserved
01h=Command is captured but not processed
02h=Command checking is in progress
03h=Command execution is in progress
04h-2Fh=Reserved
30h-3Fh=Custom
SUCCESS
00h=Reserved</td></tr>
</table>

CdbStatus		
Bit	**Attribute**	**Description**
		01h=Command completed successfully 02h=Reserved 03h=Previous CMD was ABORTED by CMD Abort 04h-1Fh=Reserved 20h-2Fh=Reserved 30h-3Fh=Custom **FAILED** 00h=Reserved 01h=CMDID unknown 02h=Parameter range error or parameter not supported 03h=Previous CMD was not properly ABORTED (by CMD Abort) 04h=Command checking time out 05h=CdbChkCode Error 06h=Password related error (command specific meaning) 07h=Command not compatible with operating status 08h-0Fh=Reserved for STS command checking error 10h-1Fh=Reserved 20h-2Fh=For individual STS command or task error 30h-3Fh=Custom
6	CdbHasFailed	Bool: CdbHasFailed bit indicates if there was a failure, after the module has completed execution of the last CDB command 0b: Last triggered CDB command completed successfully 1b: Last triggered CDB command failed
7	CdbIsBusy	Bool: CdbIsBusy status bit indicates whether the module is still busy, or idle and ready to accept a new CDB command 0b: Module idle, host can write 1b: Module busy, host needs to wait

8 CDB Event Log Management

Optical transceivers support event logging management to record critical events into non-volatile memory. They also provide a mechanism to read back the logs from memory for diagnostics and troubleshooting.

8.1 Get Event Log (CMD: 8010h, option = 0)

Use command ID **0x8010** with **option = 0** to retrieve the event logs from the module.

8.1.1 Event Log Entry Structure

Each event log entry occupies 18 bytes, consisting of 2 bytes for the UID (unique ID) and 16 bytes for informative data

- All fields are stored in little-endian byte order.
- A 16-bit UID is a sequence number that indicates the history of each event log. The smallest number is the oldest record. The UID can wrap around when it reaches U16_MAX.

Byte	Field Name	Description
0-1	UID	Unique ID as a sequence id of the log
2-5	TIME_SEC	Main timestamp of the event, in seconds (UTC).
6-7	TIME_MSEC	Fractional part of the event time, in milliseconds (UTC).
8-9	EVENT_ID	Event type identifier. (See Section 9.1.2 for details.)
10-13	DETAIL0	Extra event information, which may include technical details or troubleshooting data.
14-17	DETAIL1	Additional event information, often used for debugging or further explanation

8.1.2 Event Log IDs and Types

The following table lists supported event IDs associated with all events that shall be logged in non-volatile transceiver memory. This allows a user to unplug a module without losing all link event history which may be an essential part of network troubleshooting.

Event ID (Hex)	Event Name	Description
0x0008	MODULE _TEMPERATURE	Module temperature over threshold event
0x0009	MODULE_VOLTAGE	Module voltage over or under threshold event
0x000A	MODULE_RESET	Module reset event
0x000B	OPTICAL_POWER_ALARM	Optical power exceeds alarm threshold event
0x0102	DSP_CMD_TIMEOUT	DSP firmware command execution timeout event
0x0106	DSP_LOS	DSP loss of signal event
0x0107	DSP_LOL	DSP loss of lock event
0x0108	DSP_PREFEC_BER	DSP pre-FEC BER exceeds threshold event
0x0109	DSP_FEC_BIN	DSP FEC BIN exceeds threshold event
0x010A	DSP_MPI	Multipath interference high threshold event
0x0200	FW_UPGRADE	Module firmware upgrade event

8.1.3 Event Log Threshold

This section defines the threshold conditions for event log recording. Each event type has a specific threshold value that determines when the event should be logged in the module.

Event ID (Hex)	Event Name	Threshold Condition	Description
0x0108	DSP_PREFEC_BER	> 1e-5	Check and record when the Host side is online
0x0109	DSP_FEC_BIN	>= 8	
0x010A	DSP_MPI	> -30 (and ≠ 0) dB	Check and record when the Media side is online

8.1.4 Command Message Header (bytes 128 – 142)

- Fill the **UID** attribute (bytes 136–139) to assign the starting entry to be read; otherwise, set it to U32_MAX and the module will return the log from the oldest record
- The **Request Count** attribute (byte 140) must not exceed 6; otherwise, an error will occur. See Section 9.1.5 for details.
- If the **Target** attribute (byte 142) is set to 0, then the **Port** attribute must also be 0

CDB Eventlog Management Command Message			
Page	**Byte**	**Attribute**	**Description**
9Fh	128	CMDID	0x8010
	129		
	130	EPL length	VAR
	131		
	132	IPL length	VAR
	133	Cdb chk code	VAR
	134	RPL length	VAR
	135	RPL chk code	VAR
	136	UID	Read from oldest event log entry index, if it is 0xFFFFFFFF
	137		
	138		
	139		
	140	Request count	Request read back event log count (max 6 per command)
	141	Option	Event log management option: 0: Read event log from flash 1: Wipe all event logs and reset it to initial state

	142	End	Selection for cable end 0: local end 1: remote end
	143	Port	For breakout cases, select which remote end to retrieve the event log from. For example, for 2x200G cable, 0 for first 200G port, 1 for second 200G port. If there are no multiple remote ends, use 0.

8.1.5 Reply Message Header (bytes 136 – 244)

Once the event log management command is successfully processed by the module, the user application can retrieve the information listed below

CDB Event Log Management Reply Message			
Page	**Byte**	**Reply**	**Description**
9Fh	134	RPL length	VAR
	135	RPL chk code	VAR
	136	Request Count	Number of valid event entries returned
	137	Event Log Entry 1	Refer to section 9.1.1
	138		
	139		
	140		
	141		
	142		
	143		
	144		
	145		
	146		
	147		
	148		
	149		
	150		
	151		
	152		
	153		
	154		
	155	Event Log Entry 2	Refer to section 9.1.1
	156		
	157		
	158		

CDB Event Log Management Reply Message			
Page	**Byte**	**Reply**	**Description**
	159		
	160		
	161		
	162		
	163		
	164		
	165		
	166		
	167		
	168		
	169		
	170		
	171		
	172		
	173	Event Log Entry 3	Refer to section 9.1.1
	174		
	175		
	176		
	177		
	178		
	179		
	180		
	181		
	182		
	183		
	184		
	185		
	186		
	187		
	188		
	189		
	190		
	191	Event Log Entry 4	Refer to section 9.1.1
	192		
	193		
	194		
	195		

CDB Event Log Management Reply Message			
Page	**Byte**	**Reply**	**Description**
	196		
	197		
	198		
	199		
	200		
	201		
	202		
	203		
	204		
	205		
	206		
	207		
	208		
	209	Event Log Entry 5	Refer to section 9.1.1
	210		
	211		
	212		
	213		
	214		
	215		
	216		
	217		
	218		
	219		
	220		
	221		
	222		
	223		
	224		
	225		
	226		
	227	Event Log Entry 6	Refer to section 9.1.1
	228		
	229		
	230		
	231		
	232		

CDB Event Log Management Reply Message			
Page	**Byte**	**Reply**	**Description**
	233		
	234		
	235		
	236		
	237		
	238		
	239		
	240		
	241		
	242		
	243		
	244		
	245-255	Reserved	

8.1.6 CDB Command Status (Page 0, byte 8 and bytes 37 – 38)

These registers are updated by the module and must be read by the host after each command to determine execution status and completion

CDB Command Status				
Page	**Byte**	**Bit**	**Attribute**	**Description**
0h	8	6	CdbCmdCompleteFlag1	Latched Flag to indicate completion of a CDB command for CDB instance 1
		7	CdbCmdCompleteFlag2	Latched Flag to indicate completion of a CDB command for CDB instance 2
	37	7 - 0	CdbStatus1	Status of the most recent CDB command in CDB instance 1
	38	7 - 0	CdbStatus2	Status of the most recent CDB command in CDB instance 2

CdbStatus		
Bit	**Attribute**	**Description**
5 - 0	CdbCommandResult	The CdbCommandResult field provides more detailed classification for each of the three coarse query results that are encoded by the pair of bit 7 (CdbIsBusy) and bit 6 (CdbHasFailed)

Coarse Status	CdbIsBusy	CdbHasFailed
IN PROGRESS	1	X (don't care)

<table>
<tr><th colspan="3">CdbStatus</th></tr>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td>SUCCESS</td><td>0</td><td>0</td></tr>
<tr><td>FAILED</td><td>0</td><td>1</td></tr>
</table>

The interpretation of CdbCommandResult therefore depends on the coarse status as follows:
IN PROGRESS
00h=Reserved
01h=Command is captured but not processed
02h=Command checking is in progress
03h=Command execution is in progress
04h-2Fh=Reserved
30h-3Fh=Custom
SUCCESS
00h=Reserved
01h=Command completed successfully
02h=Reserved
03h=Previous CMD was ABORTED by CMD Abort
04h-1Fh=Reserved
20h-2Fh=Reserved
30h-3Fh=Custom
FAILED
00h=Reserved
01h=CMDID unknown
02h=Parameter range error or parameter not supported
03h=Previous CMD was not properly ABORTED (by CMD Abort)
04h=Command checking time out
05h=CdbChkCode Error
06h=Password related error (command specific meaning)
07h=Command not compatible with operating status
08h-0Fh=Reserved for STS command checking error
10h-1Fh=Reserved
20h-2Fh=For individual STS command or task error
30h-3Fh=Custom</td></tr>
<tr><td>6</td><td>CdbHasFailed</td><td>Bool: CdbHasFailed bit indicates if there was a failure, after the module has completed execution of the last CDB command
0b: Last triggered CDB command completed successfully
1b: Last triggered CDB command failed</td></tr>
<tr><td>7</td><td>CdbIsBusy</td><td>Bool: CdbIsBusy status bit indicates whether the module is still busy, or idle and ready to accept a new CDB command
0b: Module idle, host can write
1b: Module busy, host needs to wait</td></tr>
</table>

8.1.7 Event log parsing example

The screenshot below shows how an event log is parsed and displayed



ID	Time	Event	Detail1	Detail2
000001	1970-01-01 00:00:00.006	Module Reset (0x000A)	Slot 1 COLD 0x0100434C	0x64005244

8.2 Reset Event Log (CMD: 8010h, option = 1)

To wipe and reset all event logs, use the same command ID (0x8010) with option = 1. All event logs will be erased and reset to their initial state after the command is processed.

8.2.1 Command Message Header (bytes 128 – 141)

- Fill the option with 1 to reset all event log
- If the **Target** attribute (byte 142) is set to 0, then the **Port** attribute must also be 0

CDB Event Log Management Command			
Page	**Byte**	**Attribute**	**Description**
9Fh	128	CMDID	0x8010
	129		
	130	EPL length	N/A
	131		
	132	LPL length	VAR
	133	Cdb chk code	VAR
	134	RPL length	N/A
	135	RPL chk code	N/A
	136	N/A	
	137		
	138		
	139		
	140	N/A	
	141	Option	Event log management option: 0: Read event log from flash 1: Wipe all event logs and reset it to initial state
	142	Target	Select to wipe and reset all event logs for the local or remote endpoint 0: Local end 1: Remote end
	143	Port	For breakout cases, specify which remote end should be wiped and reset for all event logs. For example, for 2x200G cable, 0 for first 200G port, 1 for second 200G port. If there are no multiple remote ends, use 0.

8.3 Synchronize Event Log Time (CMD: 8011h)

This command is used to synchronize the system UTC time across the device to ensure that all event log entries are accurately timestamped.

8.3.1 Command Message Header (bytes 128 – 141)

- Fill the system current UTC time (Unix timestamp format)
- If the **Target** attribute (byte 142) is set to 0, then the **Port** attribute must also be set to 0.

CDB Event Log Management Command			
Page	**Byte**	**Attribute**	**Description**
9Fh	128	CMDID	0x8010
	129		
	130	EPL length	N/A
	131		
	132	LPL length	VAR
	133	Cdb chk code	VAR
	134	RPL length	N/A
	135	RPL chk code	N/A
	136	UTC time	Current system UTC time in Unix timestamp format (seconds since 1970-01-01 00:00:00 UTC).
	137		
	138		
	139		
	140	N/A	
	141	N/A	
	142	Target	Select local or remote endpoint for time sync. 0: Local end 1: Remote end
	143	Port	For breakout cases, specify which remote end should be time synchronized. For example, for 2x200G cable, 0 for first 200G port, 1 for second 200G port. If there are no multiple remote ends, use 0.

8.3.2 CDB Command Status (Page 0, byte 8 and bytes 37 – 38)

These registers are updated by the module and must be read by the host after each command to determine execution status and completion

CDB Command Status				
Page	**Byte**	**Bit**	**Attribute**	**Description**
0h	8	6	CdbCmdCompleteFlag1	Latched Flag to indicate completion of a CDB command for CDB instance 1
		7	CdbCmdCompleteFlag2	Latched Flag to indicate completion of a CDB command for CDB instance 2
	37	7 - 0	CdbStatus1	Status of the most recent CDB command in CDB instance 1
	38	7 - 0	CdbStatus2	Status of the most recent CDB command in CDB instance 2

<table>
<tr><th colspan="3">CdbStatus</th></tr>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>5 - 0</td><td>CdbCommandResult</td><td>The CdbCommandResult field provides more detailed classification for each of the three coarse query results that are encoded by the pair of bit 7 (CdbIsBusy) and bit 6 (CdbHasFailed)

<table>
<tr><td>Coarse Status</td><td>CdbIsBusy</td><td>CdbHasFailed</td></tr>
<tr><td>IN PROGRESS</td><td>1</td><td>X (don’t care)</td></tr>
<tr><td>SUCCESS</td><td>0</td><td>0</td></tr>
<tr><td>FAILED</td><td>0</td><td>1</td></tr>
</table>

The interpretation of CdbCommandResult therefore depends on the coarse status as follows:
IN PROGRESS
00h=Reserved
01h=Command is captured but not processed
02h=Command checking is in progress
03h=Command execution is in progress
04h-2Fh=Reserved
30h-3Fh=Custom
SUCCESS
00h=Reserved
01h=Command completed successfully
02h=Reserved
03h=Previous CMD was ABORTED by CMD Abort
04h-1Fh=Reserved
20h-2Fh=Reserved
30h-3Fh=Custom
FAILED</td></tr>
</table>

CdbStatus		
Bit	**Attribute**	**Description**
		00h=Reserved 01h=CMDID unknown 02h=Parameter range error or parameter not supported 03h=Previous CMD was not properly ABORTED (by CMD Abort) 04h=Command checking time out 05h=CdbChkCode Error 06h=Password related error (command specific meaning) 07h=Command not compatible with operating status 08h-0Fh=Reserved for STS command checking error 10h-1Fh=Reserved 20h-2Fh=For individual STS command or task error 30h-3Fh=Custom
6	CdbHasFailed	Bool: CdbHasFailed bit indicates if there was a failure, after the module has completed execution of the last CDB command 0b: Last triggered CDB command completed successfully 1b: Last triggered CDB command failed
7	CdbIsBusy	Bool: CdbIsBusy status bit indicates whether the module is still busy, or idle and ready to accept a new CDB command 0b: Module idle, host can write 1b: Module busy, host needs to wait

9 CDB Remote CMIS Register Access

Two remote CMIS access commands are supported to enable arbitrary reading from and writing to all CMIS pages from a remote endpoint on an optical link.

- Remote CMIS byte read and write (CMD: 8020h) via LPL with limited access length up to 114 bytes.
- Remote CMIS page read and write (CMD: 8021h) via EPL with full page access up to 128 bytes.

9.1 CDB Remote CMIS Byte Access (CMD: 8020h)

To partially read from or write to a remote module's CMIS registers, the host must fill in the command message as follows:

- Fill in the command arguments to select the port, register bank, register page, address, and length.
- Set the operation to 0 for read or 1 for write and set the length (maximum up to 114 bytes).
- For write operations, the data must be filled in the field starting from byte 142.

- Set the LPL length and the CDB check code.

Note on CMIS remote register:

- Writing remote byte 126 and 127 has no effect. No error will be reported.
- Reading remote byte 126 and 127 will always return remote module's byte 126 and 127. It will not return Bank and Page in command.

9.1.1 Command Message Header

CDB Byte Access Command Message			
Page	**Byte**	**Attribute**	**Description**
9Fh	128	CMDID	0x8020
	129		
	130	EPL length	N/A
	131		
	132	IPL length	VAR
	133	Cdb chk code	VAR
	134	RPL length	N/A
	135	RPL chk code	N/A
	136	Port	For breakout cases, choose which remote end to access. For example, for 2x200G cable, 0 for first 200G port, 1 for second 200G port. If there are no multiple remote ends, use 0.
	137	Bank	Select remote CMIS register bank. See byte 126. Please note remote byte 126 is not changed in the process
	138	Page	Select remote CMIS register page. See byte 127. Please note remote byte 127 is not changed in the process.
	139	Address	Select address of first CMIS register.
	140	Length	Select reading or writing length of remote registers. Valid from 1 ~ 114
	141	Operation	Select the access operation for writing to or reading from the remote end 0: read 1: write
	142-255	Data for write	The data is written to the remote end if the operation is set to 1 (write)

9.1.2 Reply Message Header

Once the remote byte access command has been successfully processed by the module, the user application can retrieve the following information to obtain the read data.

		CDB Byte Access Reply Message	
Page	**Byte**	**Attribute**	**Description**
9Fh	134	RPL length	VAR
	135	RPL chk code	VAR
	136-141	Reserved	
	142-255	Data for read	The data is read from the remote end if the operation is set to 0 (read)

9.1.3 CDB Command Status (Page 0, byte 8 and bytes 37 – 38)

These registers are updated by the module and must be read by the host after each command to determine execution status and completion

CDB Command Status				
Page	**Byte**	**Bit**	**Attribute**	**Description**
0h	8	6	CdbCmdCompleteFlag1	Latched Flag to indicate completion of a CDB command for CDB instance 1
		7	CdbCmdCompleteFlag2	Latched Flag to indicate completion of a CDB command for CDB instance 2
	37	7 - 0	CdbStatus1	Status of the most recent CDB command in CDB instance 1
	38	7 - 0	CdbStatus2	Status of the most recent CDB command in CDB instance 2

CdbStatus		
Bit	**Attribute**	**Description**
5 - 0	CdbCommandResult	The CdbCommandResult field provides more detailed classification for each of the three coarse query results that are encoded by the pair of bit 7 (CdbIsBusy) and bit 6 (CdbHasFailed)

Coarse Status	CdbIsBusy	CdbHasFailed
IN PROGRESS	1	X (don't care)
SUCCESS	0	0
FAILED	0	1

CdbStatus		
Bit	**Attribute**	**Description**
		The interpretation of CdbCommandResult therefore depends on the coarse status as follows: **IN PROGRESS** 00h=Reserved 01h=Command is captured but not processed 02h=Command checking is in progress 03h=Command execution is in progress 04h-2Fh=Reserved 30h-3Fh=Custom **SUCCESS** 00h=Reserved 01h=Command completed successfully 02h=Reserved 03h=Previous CMD was ABORTED by CMD Abort 04h-1Fh=Reserved 20h-2Fh=Reserved 30h-3Fh=Custom **FAILED** 00h=Reserved 01h=CMDID unknown 02h=Parameter range error or parameter not supported 03h=Previous CMD was not properly ABORTED (by CMD Abort) 04h=Command checking time out 05h=CdbChkCode Error 06h=Password related error (command specific meaning) 07h=Command not compatible with operating status 08h-0Fh=Reserved for STS command checking error 10h-1Fh=Reserved 20h-2Fh=For individual STS command or task error 30h-3Fh=Custom
6	CdbHasFailed	Bool: CdbHasFailed bit indicates if there was a failure, after the module has completed execution of the last CDB command 0b: Last triggered CDB command completed successfully 1b: Last triggered CDB command failed
7	CdbIsBusy	Bool: CdbIsBusy status bit indicates whether the module is still busy, or idle and ready to accept a new CDB command 0b: Module idle, host can write 1b: Module busy, host needs to wait

9.2 CDB Remote CMIS Page Access (CMD: 8021h)

To access the entire CMIS page registers, the user application must fill in the message for the page access command

- Fill in the port, bank, page, address in the command message.
- Set the operation to 0 for read or 1 for write and set the length and corresponding EPL length, 256 for page 0, 128 for other pages.
- Read/Write data are managed starting from the byte 0x80 of page 0xA0.
- To write the entire page registers, the user must fill in the EPL pages before sending the command message.
- For read the entire page, the user can read the EPL pages after the command message has been successfully processed by the module.

9.2.1 Command Message Header

CDB Page Access Command Message			
Page	**Byte**	**Attribute**	**Description**
9Fh	128	CMDID	0x8021
	129		
	130	EPL length	256: for page 0 128: for page 1 ~ 255
	131		
	132	IPL length	VAR
	133	Cdb chk code	VAR
	134	RPL length	N/A
	135	RPL chk code	N/A
	136	Port	For breakout cases, choose which remote end to access. For example, for 2x200G cable, 0 for first 200G port, 1 for second 200G port. If there are no multiple remote ends, use 0.
	137	Bank	Select remote CMIS register bank. See byte 126. Please note remote byte 126 is not changed in the process
	138	Page	Select remote CMIS register page. See byte 127. Please note remote byte 127 is not changed in the process.
	139	EPL chk code	One's complement of the sum of all data to be written in pages 0xA0 and 0xA1. Please note that only page 0 uses both 0xA0 and 0xA1 to carry lower and upper page 0 data; for other pages, only 0xA0 is used.

CDB Page Access Command Message			
Page	**Byte**	**Attribute**	**Description**
	140	Operation	Select the access operation for writing to or reading from the remote end 0: read 1: write

9.2.2 Reply Message Header

Once the remote byte access command has been successfully processed by the module, the user application can retrieve the following information to obtain the read data.

CDB Page Access Reply Message			
Page	**Byte**	**Attribute**	**Description**
9Fh	134	RPL length	VAR
	135	RPL chk code	VAR
	136-255	Reserved	

9.2.3 CDB Command Status (Page 0, byte 8 and bytes 37 – 38)

These registers are updated by the module and must be read by the host after each command to determine execution status and completion

CDB Command Status				
Page	**Byte**	**Bit**	**Attribute**	**Description**
0h	8	6	CdbCmdCompleteFlag1	Latched Flag to indicate completion of a CDB command for CDB instance 1
		7	CdbCmdCompleteFlag2	Latched Flag to indicate completion of a CDB command for CDB instance 2
	37	7 - 0	CdbStatus1	Status of the most recent CDB command in CDB instance 1
	38	7 - 0	CdbStatus2	Status of the most recent CDB command in CDB instance 2

CdbStatus		
Bit	**Attribute**	**Description**
5 - 0	CdbCommandResult	The CdbCommandResult field provides more detailed classification for each of the three coarse query results that are encoded by the pair of bit 7 (CdbIsBusy) and bit 6 (CdbHasFailed)

<table>
<tr><th colspan="3">CdbStatus</th></tr>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td>Coarse Status</td><td>CdbIsBusy</td><td>CdbHasFailed</td></tr>
<tr><td>IN PROGRESS</td><td>1</td><td>X (don’t care)</td></tr>
<tr><td>SUCCESS</td><td>0</td><td>0</td></tr>
<tr><td>FAILED</td><td>0</td><td>1</td></tr>
</table>
The interpretation of CdbCommandResult therefore depends on the coarse status as follows:
IN PROGRESS
00h=Reserved
01h=Command is captured but not processed
02h=Command checking is in progress
03h=Command execution is in progress
04h-2Fh=Reserved
30h-3Fh=Custom
SUCCESS
00h=Reserved
01h=Command completed successfully
02h=Reserved
03h=Previous CMD was ABORTED by CMD Abort
04h-1Fh=Reserved
20h-2Fh=Reserved
30h-3Fh=Custom
FAILED
00h=Reserved
01h=CMDID unknown
02h=Parameter range error or parameter not supported
03h=Previous CMD was not properly ABORTED (by CMD Abort)
04h=Command checking time out
05h=CdbChkCode Error
06h=Password related error (command specific meaning)
07h=Command not compatible with operating status
08h-0Fh=Reserved for STS command checking error
10h-1Fh=Reserved
20h-2Fh=For individual STS command or task error
30h-3Fh=Custom
</td></tr>
<tr><td>6</td><td>CdbHasFailed</td><td>Bool: CdbHasFailed bit indicates if there was a failure, after the module has completed execution of the last CDB command
0b: Last triggered CDB command completed successfully
1b: Last triggered CDB command failed</td></tr>
<tr><td>7</td><td>CdbIsBusy</td><td>Bool: CdbIsBusy status bit indicates whether the module is still busy, or idle and ready to accept a new CDB command
0b: Module idle, host can write</td></tr>
</table>

CdbStatus		
Bit	**Attribute**	**Description**
		1b: Module busy, host needs to wait

10 CDB Threshold Control (CMD: 8030h)

To be defined as part of workstream efforts.

11 CDB Firmware Management Features

Compliant optical transceivers shall support two methods for determining available firmware management features. The first method shall be the standard procedure defined in CMIS. The second method shall be this custom protocol that enables remote queries through in-band communication from one end to the opposite end.

11.1 CDB Firmware Management Features (Local)

Issue the CMD 0041h command to query which firmware management features are supported.

Detailed information can be found in Sections 9.4.2 of CMIS v5.3.

11.2 CDB Firmware Management Features (Remote)

Compliant transceivers shall support vendor-defined commands to remotely query which firmware management features are supported.

Remote firmware management features command uses a vendor-defined command ID range to distinguish operations targeting different ports. The command ID format follows a structured encoding scheme:

Remote FW management features command ID format		
Bit	**Define**	**Description**
15	Remote access	Always set to 1 to indicate a remote FW management feature command
14	Port number	The port is used for breakout cases to select which remote end to access. For example, in a 2x200G cable, value 0 selects the first 200G port, and value 1 selects the second 200G port
13		
12		
11 - 0	FW management features command	Retain the CMIS FW management features command ID format (041h)

11.3 CDB Firmware Management Features (CMD: 8041h to F041h)

Issue the command to remotely query which firmware management features are supported

CDB Remote Firmware Management Features Command Message			
Page	**Byte**	**Attribute**	**Description**
9Fh	128	CMDID	Starting from 0x8041 for Port 0 (bits [14:12] = 0), each port increments the 3-bit port index in bits [14:12], resulting in 0x9041 for Port 1, 0xA041 for Port 2, and so forth
	129		
	130	EPL length	N/A
	131		
	132	IPL length	VAR
	133	Cdb chk code	VAR
	134	RPL length	N/A
	135	RPL chk code	N/A

11.3.1 Reply Message Header

Once the remote firmware management features command is successfully processed by the module, the user application can retrieve the information listed below.

CDB Remote Firmware Management Features Reply Message			
Page	**Byte**	**Attribute**	**Description**
9Fh	134	RPL length	VAR
	135	RPL chk code	VAR
	136		Reserved
	137	ImageReadback MaxDurationCoding SkippingErasedBlocks CopyCmd AbortCmd	bit7: ImageReadback 0b = Full Image Readback Not Supported 1b = Full Image Readback Supported (see CMIS v5.3 section 7.3.1) bit6-4: Reserved bit3: MaxDurationCoding 0b = max duration multiplier M is 1 1b = max duration multiplier M is 10 bit2: SkippingErasedBlocks 0b = Skipping erased blocks Not Supported 1b = Skipping erased blocks Supported bit1: CopyCmd 0b = CMD x108h (Copy image) Not Supported

CDB Remote Firmware Management Features Reply Message			
Page	**Byte**	**Attribute**	**Description**
			1b = CMD x108h (Copy image) Supported bit0: AbortCmd 0b = CMD x102h (Abort) Not Supported 1b = CMD x102h (Abort) Supported
	138	StartCmdPayloadSize	This defines the number of bytes that the host must extract from the beginning of the vendor-delivered binary firmware image file and send to the module in CMD x101h (Start).
	139	ErasedByte	This is the value representing an erased byte. The purpose of advertising this byte is to optionally reduce download time by allowing the host to skip sending blocks of the image containing ErasedByte values only.
	140	ReadWriteLengthExt	ReadWriteLengthExt = **i** specifies the allowable **additional** number of byte octets in a READ or a WRITE, specifically for **Firmware Management Commands** (IDs x100h-x1FFh) as follows **EPL:** For accessing the multi-page **EPL** field, the allowable length extension is i byte octets (8 bytes). **LPL**: For accessing the **LPL** field on page **9Fh**, the allowable length extension is min (**i**, 15) byte octets. This leads to the maximum length of a READ or a WRITE **Value Maximum Number of Bytes (EPL)** 0: 8 bytes (no extension of general length limit) i: 8 * (1+i) bytes (0 ≤ i ≤ 255) 255: 8 * 256 = 2048 bytes **Value Maximum Number of Bytes (LPL)** 0: 8 bytes (no extension of general length limit) i: 8 * (1+i) bytes (0 ≤ i ≤ 15) i: 8 * 16 = 128 bytes (16 ≤ i ≤ 256)
	141	WriteMechanism	Firmware update supported mechanism. 00h: None Supported. 01h: Write to LPL supported. 10h: Write to EPL supported. 11h: Both Write to LPL and EPL supported.
	142	ReadMechanism	Firmware read / readback support mechanism. 00h: None Supported. 01h: Read via LPL supported.

CDB Remote Firmware Management Features Reply Message			
Page	**Byte**	**Attribute**	**Description**
			10h: Read via EPL supported. 11h: Both Read via LPL and EPL supported.
	143	HitlessRestart	0: CMD Run Image causes a reset. Traffic is affected. 1: CMD Run Image may reset but module will do its best to maintain traffic and management states. Data path functions are not reset.
	144	MaxDurationStart	U16 Maximum time in M ms for a CDB Start command to complete execution, where M is defined by bit 137.3
	145		
	146	MaxDurationAbort	U16 Maximum time in M ms for a CDB Abort command to complete execution, where M is defined by bit 137.3
	147		
	148	MaxDurationWrite	U16 Maximum time in M ms for a CDB Write command to complete execution, where M is defined by bit 137.3
	149		
	150	MaxDurationComplete	U16 Maximum time in M ms for a CDB Complete command to complete execution, M defined by bit 137.3
	151		
	152	MaxDurationCopy	U16 Maximum time in M ms for a CDB Copy command to complete execution, where M is defined by bit 137.3
	153		

11.3.2 CDB Command Status (Page 0, byte 8 and bytes 37 – 38)

These registers are updated by the module and must be read by the host after each command to determine execution status and completion.

CDB Command Status				
Page	**Byte**	**Bit**	**Attribute**	**Description**
0h	8	6	CdbCmdCompleteFlag1	Latched Flag to indicate completion of a CDB command for CDB instance 1
		7	CdbCmdCompleteFlag2	Latched Flag to indicate completion of a CDB command for CDB instance 2
	37	7 - 0	CdbStatus1	Status of the most recent CDB command in CDB instance 1
	38	7 - 0	CdbStatus2	Status of the most recent CDB command in CDB instance 2

<table>
<tr><th colspan="3">CdbStatus</th></tr>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>5 – 0</td><td>CdbCommandResult</td><td>The CdbCommandResult field provides more detailed classification for each of the three coarse query results that are encoded by the pair of bit 7 (CdbIsBusy) and bit 6 (CdbHasFailed)

<table>
<tr><td>Coarse Status</td><td>CdbIsBusy</td><td>CdbHasFailed</td></tr>
<tr><td>IN PROGRESS</td><td>1</td><td>X (don’t care)</td></tr>
<tr><td>SUCCESS</td><td>0</td><td>0</td></tr>
<tr><td>FAILED</td><td>0</td><td>1</td></tr>
</table>

The interpretation of CdbCommandResult therefore depends on the coarse status as follows:
IN PROGRESS
00h=Reserved
01h=Command is captured but not processed
02h=Command checking is in progress
03h=Command execution is in progress
04h-2Fh=Reserved
30h-3Fh=Custom
SUCCESS
00h=Reserved
01h=Command completed successfully
02h=Reserved
03h=Previous CMD was ABORTED by CMD Abort
04h-1Fh=Reserved
20h-2Fh=Reserved
30h-3Fh=Custom
FAILED
00h=Reserved
01h=CMDID unknown
02h=Parameter range error or parameter not supported
03h=Previous CMD was not properly ABORTED (by CMD Abort)
04h=Command checking time out
05h=CdbChkCode Error
06h=Password related error (command specific meaning)
07h=Command not compatible with operating status
08h-0Fh=Reserved for STS command checking error
10h-1Fh=Reserved
20h-2Fh=For individual STS command or task error
30h-3Fh=Custom</td></tr>
<tr><td>6</td><td>CdbHasFailed</td><td>Bool: CdbHasFailed bit indicates if there was a failure, after the module has completed execution of the last CDB command
0b: Last triggered CDB command completed successfully
1b: Last triggered CDB command failed</td></tr>
</table>

CdbStatus		
Bit	**Attribute**	**Description**
7	CdbIsBusy	Bool: CdbIsBusy status bit indicates whether the module is still busy, or idle and ready to accept a new CDB command 0b: Module idle, host can write 1b: Module busy, host needs to wait

12 CDB Firmware Management

Compliant optical transceivers shall support two CDB firmware upgrade methods. The first method shall be the standard procedure defined in CMIS. The second method shall be a custom protocol that enables remote queries through in-band communication from one end to the opposite end.

12.1 CDB Firmware Upgrade (Local)

The CDB local firmware upgrade flow is a standard CMIS process used to update the firmware of a device via Common Data Block (CDB). This process ensures that the firmware is reliably written to the device in blocks, with verification and error status at each step. The essential steps involve initializing the upgrade mode, transferring firmware data in blocks, and finalizing the update to ensure data integrity.

Detailed information can be found in Sections 7.3.1.2 and 9.2 of CMIS v5.3.

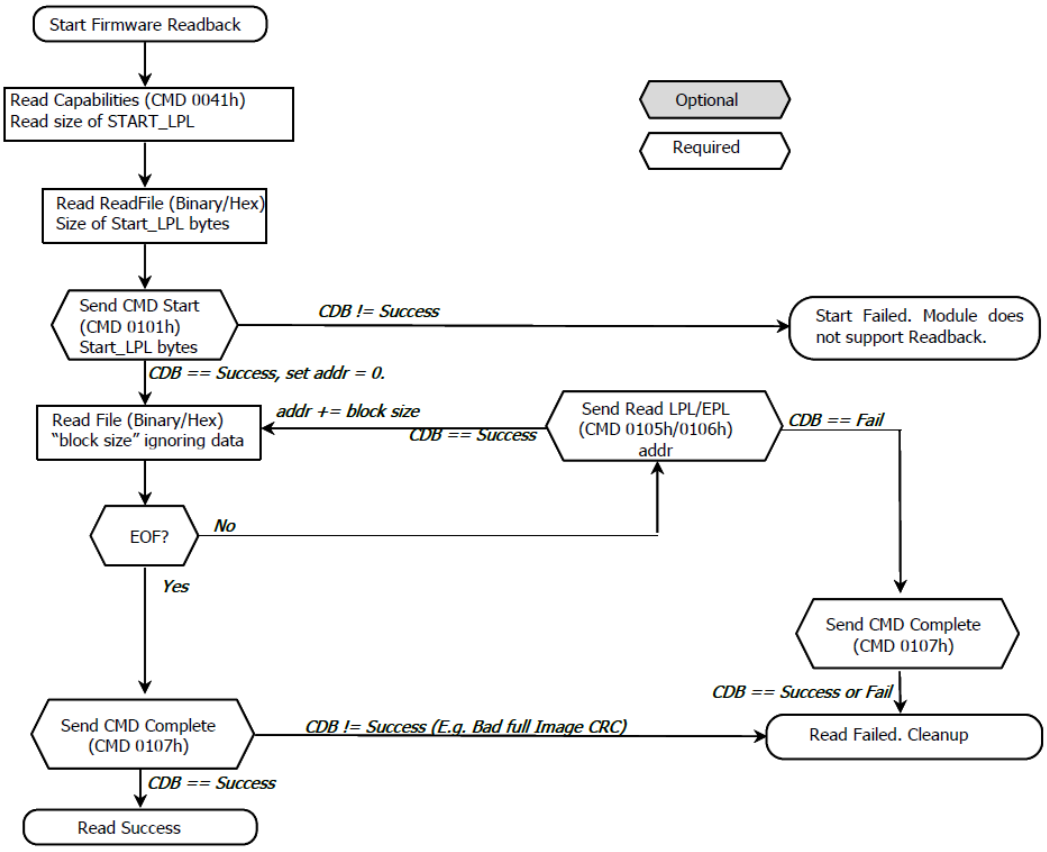


Figure 3: Firmware Upload Using CDB

12.1.1 Get Firmware information (CMD: 0100h)

Issue the CMD 0100h command to get the firmware versions and the default running firmware images that reside in the module

12.1.2 Start Firmware Download (CMD: 0101h)

Issue the CMD 0101h (Start) command with the START_LPL bytes to initiate the firmware upgrade mode.

- If this command fails, the process is aborted

12.1.3 Abort Firmware Download (CMD: 0102h)

Aborts the firmware update process.

- If this command fails, the process is aborted

12.1.4 Firmware Data Transfer via LPL/EPL (CMD: 0103h/0104h)

For each block of firmware data:

- If the block data is not equal to the erase byte, send CMD Write EPL (CMD 0104h) to write the block to the device.
- If the write command fails, abort the process and perform cleanup.
- Continue until the end of the firmware file (EOF) is reached

12.1.5 Complete Firmware Download (CMD: 0107h)

After all blocks are written, send CMD 0107h (Complete) to finalize the firmware update.

- If this command fails (e.g., due to a bad image CRC), the update is considered failed, and cleanup is performed.
- If successful, the firmware update is complete.

12.1.6 Run Firmware Image (CMD: 0109h)

Issue CMD 0109h to instruct the device to run the newly downloaded firmware image from its internal firmware banks.

- This step may be used to switch to the new firmware version after an update.

12.1.7 Commit Firmware Image (CMD: 010Ah)

Once the new firmware is running and verified to be working properly, issue CMD 010Ah to commit the running image.

- This sets the new firmware as the default image to be used after any future resets.

12.2 CDB Firmware Upgrade (Remote)

In addition to the standard CDB firmware upgrade, compliant transceivers shall support vendor-defined remote firmware upgrade commands. This feature enables firmware binaries to be upgraded from the host (load side) to the remote end of the optical link. The upgrade process closely follows the standard CDB flow but uses a different set of command IDs specific to remote operations.

Remote firmware management commands use a vendor-defined command ID range to distinguish operations targeting different ports. The command ID format follows a structured encoding scheme.

Remote FW management command ID format		
Bit	**Define**	**Description**

15	Remote access	Always set to 1 to indicate a remote FW management command
14	Port number	The port is used for split cables to select which remote end to access. For example, in a 2x200G cable, value 0 selects the first 200G port, and value 1 selects the second 200G port
13		
12		
11 - 0	FW management command	Retain the CMIS FW management command ID format (100h to 11Fh)

To perform a remote upgrade, a recommended upgrade flow using the commands shown in the block diagram below is introduced.

- CMD 8100h: Get firmware information.
- CMD 8101h: Start remote firmware download.
- CMD 8104h: Transfer firmware data blocks to the remote end.
- CMD 8107h: Complete the remote firmware download process.
- CMD 8109h: Run the remote firmware image.
- CMD 810Ah: Commit the remote firmware image as the default for future resets.

Note: The command IDs listed above are based on Port 0 (bits [14:12] = 0). For other ports, the same command set applies with the corresponding port number encoded in bits [14:12]. For example, Port 1 uses command IDs in the range 0x9100 to 0x910A, Port 2 uses 0xA100 to 0xA10A, and so on.

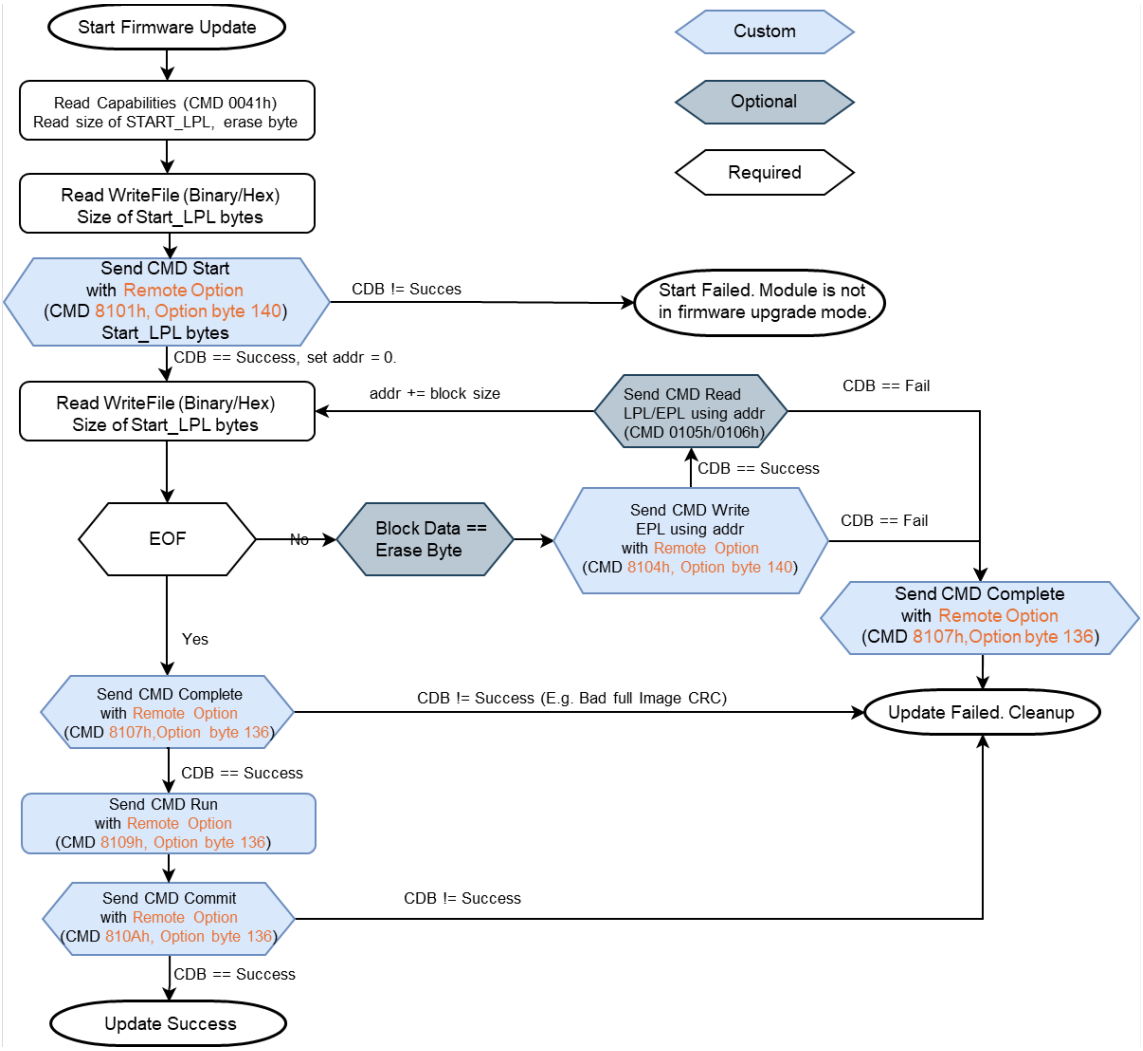


Figure 4: Remote Firmware Update Using Customized CDB Command

12.2.1 Get Firmware information (CMD: 8100h to F100h)

Compliant optical transceivers support vendor-defined commands to remotely query firmware version and status information, in addition to the standard CDB firmware information retrieval.

Issue the command to remotely query the firmware versions and default running firmware images present in the module

CDB Remote Get FW Info Command Message			
Page	**Byte**	**Attribute**	**Description**
9Fh	128	CMDID	

CDB Remote Get FW Info Command Message			
Page	**Byte**	**Attribute**	**Description**
	129		Starting from 0x8100 for Port 0 (bits [14:12] = 0), each port increments the 3-bit port index in bits [14:12], resulting in 0x9100 for Port 1, 0xA100 for Port 2, and so forth
	130	EPL length	N/A
	131		
	132	IPL length	VAR
	133	Cdb chk code	VAR
	134	RPL length	N/A
	135	RPL chk code	N/A

12.2.2 Reply Message Header

Once the remote byte access command has been successfully processed by the module, the user application can retrieve the following information to obtain the read data.

CDB Get FW Info Reply Message			
Page	**Byte**	**Attribute**	**Description**
9Fh	134	RPL length	VAR
	135	RPL chk code	VAR
	136	FirmwareStatus	Bitmask to indicate FW Status. Image in Bank A: Bit 0: Operational Status Bit 1: Administrative Status Bit 2: Validity Status Bit 3: Reserved Image in Bank B: Bit 4: Operational Status Bit 5: Administrative Status Bit 6: Validity Status Bit 7: Reserved Encoding as follows: Operational Status: 1 = running, 0 = not running Administrative Status: 1=committed, 0=uncommitted Validity Status: 1 = invalid, 0 = valid Note: Zero-encoding of valid maintains backwards compatibility with CMIS 4.0

CDB Get FW Info Reply Message			
Page	**Byte**	**Attribute**	**Description**
			Hints: 0x00h Factory image is running (if supported) See also section 7.3.1.4 for a more detailed description.
	137	ImageInformation	Bit 0: Firmware image A information in 9Fh:138-173 Bit 1: Firmware image B information in 9Fh:174-209 Bit 2: Factory or Boot image information in 9Fh:201-245
	138	ImageAMajor	Image A firmware major revision
	139	ImageAMinor	Image A firmware minor revision
	140	ImageABuild	Image A firmware build number
	141		
	142 - 173	ImageAExtraString	Additional information
	174	ImageBMajor	Image B firmware major revision
	175	ImageBMinor	Image B firmware minor revision
	176	ImageBBuild	Image B firmware build number
	177		
	178-209	ImageBExtraString	Additional information
	210	FactoryBootMajor	Factory or Boot image firmware major revision
	211	FactoryBootMinor	Factory or Boot image firmware minor revision

CDB Get FW Info Reply Message			
Page	**Byte**	**Attribute**	**Description**
	212	FactoryBootBuild	Factory or Boot image firmware build number
	213		
	214-245	FactoryBootExtraStr	Additional information

12.2.3 CDB Command Status (Page 0, byte 8 and bytes 37 – 38)

These registers are updated by the module and must be read by the host after each command to determine execution status and completion.

CDB Command Status				
Page	**Byte**	**Bit**	**Attribute**	**Description**
0h	8	6	CdbCmdCompleteFlag1	Latched Flag to indicate completion of a CDB command for CDB instance 1
		7	CdbCmdCompleteFlag2	Latched Flag to indicate completion of a CDB command for CDB instance 2
	37	7 - 0	CdbStatus1	Status of the most recent CDB command in CDB instance 1
	38	7 - 0	CdbStatus2	Status of the most recent CDB command in CDB instance 2

CdbStatus		
Bit	**Attribute**	**Description**
5 – 0	CdbCommandResult	The CdbCommandResult field provides more detailed classification for each of the three coarse query results that are encoded by the pair of bit 7 (CdbIsBusy) and bit 6 (CdbHasFailed) Coarse Status / CdbIsBusy / CdbHasFailed **IN PROGRESS** / 1 / X (don't care) **SUCCESS** / 0 / 0 **FAILED** / 0 / 1 The interpretation of CdbCommandResult therefore depends on the coarse status as follows: **IN PROGRESS** 00h=Reserved 01h=Command is captured but not processed 02h=Command checking is in progress 03h=Command execution is in progress 04h-2Fh=Reserved

CdbStatus		
Bit	**Attribute**	**Description**
		30h-3Fh=Custom **SUCCESS** 00h=Reserved 01h=Command completed successfully 02h=Reserved 03h=Previous CMD was ABORTED by CMD Abort 04h-1Fh=Reserved 20h-2Fh=Reserved 30h-3Fh=Custom **FAILED** 00h=Reserved 01h=CMDID unknown 02h=Parameter range error or parameter not supported 03h=Previous CMD was not properly ABORTED (by CMD Abort) 04h=Command checking time out 05h=CdbChkCode Error 06h=Password related error (command specific meaning) 07h=Command not compatible with operating status 08h-0Fh=Reserved for STS command checking error 10h-1Fh=Reserved 20h-2Fh=For individual STS command or task error 30h-3Fh=Custom
6	CdbHasFailed	Bool: CdbHasFailed bit indicates if there was a failure, after the module has completed execution of the last CDB command 0b: Last triggered CDB command completed successfully 1b: Last triggered CDB command failed
7	CdbIsBusy	Bool: CdbIsBusy status bit indicates whether the module is still busy, or idle and ready to accept a new CDB command 0b: Module idle, host can write 1b: Module busy, host needs to wait

12.3 Start Remote Firmware Download (CMD: 8101h to F101h)

Initiate the remote firmware upgrade process by sending this command to the module. If this command fails, the process is aborted.

CDB Remote Start FW Download Command Message Header			
Page	**Byte**	**Attribute**	**Description**
9Fh	128	CMDID	

CDB Remote Start FW Download Command Message Header			
Page	**Byte**	**Attribute**	**Description**
	129		Starting from 0x8101 for Port 0 (bits [14:12] = 0), each port increments the 3-bit port index in bits [14:12], resulting in 0x9101 for Port 1, 0xA101 for Port 2, and so forth
	130	EPL length	N/A
	131		
	132	IPL length	VAR
	133	Cdb chk code	VAR
	134	RPL length	N/A
	135	RPL chk code	N/A
	136	Image Size	U32 Size of firmware image to download into the module. This should be the file size including the LPL bytes sent as vendor data in this message.
	137		
	138		
	139		
	140-143	Reserved	Reserved
	144-255	Vendor Data	The vendor may send up to 112 bytes of information in the Start Firmware Download command. It is recommended that the binary file delivered has up to 112 bytes of header that is sent to the module. The information within this field can e.g. be used by a vendor to reject an incorrect file (binary file) and prevent firmware loading of an incorrect file presented to the module.

12.3.1 Reply Message Header

CDB Remote Start FW Download Command (Reply)			
Page	**Byte**	**Reply**	**Description**
9Fh	134	RPL length	VAR
	135	RPL chk code	VAR

12.3.2 CDB Command Status (Page 0, byte 8 and bytes 37 – 38)

These registers are updated by the module and must be read by the host after each command to determine execution status and completion

CDB Command Status				
Page	**Byte**	**Bit**	**Attribute**	**Description**
0h	8	6	CdbCmdCompleteFlag1	Latched Flag to indicate completion of a CDB command for CDB instance 1
		7	CdbCmdCompleteFlag2	Latched Flag to indicate completion of a CDB command for CDB instance 2
	37	7 - 0	CdbStatus1	Status of the most recent CDB command in CDB instance 1
	38	7 - 0	CdbStatus2	Status of the most recent CDB command in CDB instance 2

<table>
<tr><th colspan="3">CdbStatus</th></tr>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>5 – 0</td><td>CdbCommandResult</td><td>The CdbCommandResult field provides more detailed classification for each of the three coarse query results that are encoded by the pair of bit 7 (CdbIsBusy) and bit 6 (CdbHasFailed)

<table>
<tr><td>Coarse Status</td><td>CdbIsBusy</td><td>CdbHasFailed</td></tr>
<tr><td>IN PROGRESS</td><td>1</td><td>X (don’t care)</td></tr>
<tr><td>SUCCESS</td><td>0</td><td>0</td></tr>
<tr><td>FAILED</td><td>0</td><td>1</td></tr>
</table>

The interpretation of CdbCommandResult therefore depends on the coarse status as follows:
IN PROGRESS
00h=Reserved
01h=Command is captured but not processed
02h=Command checking is in progress
03h=Command execution is in progress
04h-2Fh=Reserved
30h-3Fh=Custom
SUCCESS
00h=Reserved
01h=Command completed successfully
02h=Reserved
03h=Previous CMD was ABORTED by CMD Abort
04h-1Fh=Reserved
20h-2Fh=Reserved
30h-3Fh=Custom
FAILED
00h=Reserved
01h=CMDID unknown
02h=Parameter range error or parameter not supported</td></tr>
</table>

CdbStatus		
Bit	**Attribute**	**Description**
		03h=Previous CMD was not properly ABORTED (by CMD Abort) 04h=Command checking time out 05h=CdbChkCode Error 06h=Password related error (command specific meaning) 07h=Command not compatible with operating status 08h-0Fh=Reserved for STS command checking error 10h-1Fh=Reserved 20h-2Fh=For individual STS command or task error 30h-3Fh=Custom
6	CdbHasFailed	Bool: CdbHasFailed bit indicates if there was a failure, after the module has completed execution of the last CDB command 0b: Last triggered CDB command completed successfully 1b: Last triggered CDB command failed
7	CdbIsBusy	Bool: CdbIsBusy status bit indicates whether the module is still busy, or idle and ready to accept a new CDB command 0b: Module idle, host can write 1b: Module busy, host needs to wait

12.4 Abort Remote Firmware Download (CMD: 8102h to F102h)

Abort the remote firmware upgrade process by sending this command to the module. If this command fails, the process is aborted.

		CDB Remote Abort FW Download Command Message Header	
Page	**Byte**	**Attribute**	**Description**
9Fh	128	CMDID	Starting from 0x8102 for Port 0 (bits [14:12] = 0), each port increments the 3-bit port index in bits [14:12], resulting in 0x9102 for Port 1, 0xA102 for Port 2, and so forth
	129		
	130	EPL length	N/A
	131		
	132	IPL length	VAR
	133	Cdb chk code	VAR
	134	RPL length	N/A
	135	RPL chk code	N/A

12.4.1 Reply Message Header

CDB Remote Abort FW Download Command (Reply)			
Page	**Byte**	**Reply**	**Description**
9Fh	134	RPL length	VAR
	135	RPL chk code	VAR

12.4.2 CDB Command Status (Page 0, byte 8 and bytes 37 – 38)

These registers are updated by the module and must be read by the host after each command to determine execution status and completion

CDB Command Status				
Page	**Byte**	**Bit**	**Attribute**	**Description**
0h	8	6	CdbCmdCompleteFlag1	Latched Flag to indicate completion of a CDB command for CDB instance 1
		7	CdbCmdCompleteFlag2	Latched Flag to indicate completion of a CDB command for CDB instance 2
	37	7 - 0	CdbStatus1	Status of the most recent CDB command in CDB instance 1
	38	7 - 0	CdbStatus2	Status of the most recent CDB command in CDB instance 2

CdbStatus		
Bit	**Attribute**	**Description**
5 – 0	CdbCommandResult	The CdbCommandResult field provides more detailed classification for each of the three coarse query results that are encoded by the pair of bit 7 (CdbIsBusy) and bit 6 (CdbHasFailed) Coarse Status / CdbIsBusy / CdbHasFailed **IN PROGRESS** / 1 / X (don't care) **SUCCESS** / 0 / 0 **FAILED** / 0 / 1 The interpretation of CdbCommandResult therefore depends on the coarse status as follows: **IN PROGRESS** 00h=Reserved 01h=Command is captured but not processed 02h=Command checking is in progress 03h=Command execution is in progress 04h-2Fh=Reserved 30h-3Fh=Custom

CdbStatus		
Bit	**Attribute**	**Description**
		SUCCESS 00h=Reserved 01h=Command completed successfully 02h=Reserved 03h=Previous CMD was ABORTED by CMD Abort 04h-1Fh=Reserved 20h-2Fh=Reserved 30h-3Fh=Custom **FAILED** 00h=Reserved 01h=CMDID unknown 02h=Parameter range error or parameter not supported 03h=Previous CMD was not properly ABORTED (by CMD Abort) 04h=Command checking time out 05h=CdbChkCode Error 06h=Password related error (command specific meaning) 07h=Command not compatible with operating status 08h-0Fh=Reserved for STS command checking error 10h-1Fh=Reserved 20h-2Fh=For individual STS command or task error 30h-3Fh=Custom
6	CdbHasFailed	Bool: CdbHasFailed bit indicates if there was a failure, after the module has completed execution of the last CDB command 0b: Last triggered CDB command completed successfully 1b: Last triggered CDB command failed
7	CdbIsBusy	Bool: CdbIsBusy status bit indicates whether the module is still busy, or idle and ready to accept a new CDB command 0b: Module idle, host can write 1b: Module busy, host needs to wait

12.5 Remote Firmware Data Transfer (CMD: 8104h to F104h)

Transfer the firmware image to the remote end in blocks.

- For each block, send the appropriate command to write the data.
- If any block transfer fails, abort the process and perform cleanup.
- Continue until the entire firmware image has been transferred.

CDB Remote FW Data Transfer Command Message			
Page	**Byte**	**Attribute**	**Description**
9Fh	128	CMDID	

CDB Remote FW Data Transfer Command Message			
Page	**Byte**	**Attribute**	**Description**
	129		Starting from 0x8104 for Port 0 (bits [14:12] = 0), each port increments the 3-bit port index in bits [14:12], resulting in 0x9104 for Port 1, 0xA104 for Port 2, and so forth
	130	EPL length	N/A
	131		
	132	IPL length	VAR
	133	Cdb chk code	VAR
	134	RPL length	N/A
	135	RPL chk code	N/A
	136	BlockAddress	U32 Starting byte address of this block of data within the supplied image file minus the size of the “Start Command Payload Size”. See section 7.3.1.2.
	137		
	138		
	139		

Command Data (EPL)			
Page	**Byte**	**Attribute**	**Description**
A0h-AFh	128-255	FirmwareBlock	Firmware image data

12.5.1 Reply Message Header

CDB Remote Start FW Download Command (Reply)			
Page	**Byte**	**Reply**	**Description**
9Fh	134	RPL length	VAR
	135	RPL chk code	VAR

12.5.2 CDB Command Status (Page 0, byte 8 and bytes 37 – 38)

These registers are updated by the module and must be read by the host after each command to determine execution status and completion

CDB Command Status				
Page	**Byte**	**Bit**	**Attribute**	**Description**
0h	8	6	CdbCmdCompleteFlag1	Latched Flag to indicate completion of a CDB command for CDB instance 1
		7	CdbCmdCompleteFlag2	Latched Flag to indicate completion of a CDB command for CDB instance 2

CDB Command Status				
Page	**Byte**	**Bit**	**Attribute**	**Description**
	37	7 - 0	CdbStatus1	Status of the most recent CDB command in CDB instance 1
	38	7 - 0	CdbStatus2	Status of the most recent CDB command in CDB instance 2

<table>
<tr><th colspan="3">CdbStatus</th></tr>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>5 – 0</td><td>CdbCommandResult</td><td>
The CdbCommandResult field provides more detailed classification for each of the three coarse query results that are encoded by the pair of bit 7 (CdbIsBusy) and bit 6 (CdbHasFailed)

<table>
<tr><td>Coarse Status</td><td>CdbIsBusy</td><td>CdbHasFailed</td></tr>
<tr><td>IN PROGRESS</td><td>1</td><td>X (don’t care)</td></tr>
<tr><td>SUCCESS</td><td>0</td><td>0</td></tr>
<tr><td>FAILED</td><td>0</td><td>1</td></tr>
</table>

The interpretation of CdbCommandResult therefore depends on the coarse status as follows:
IN PROGRESS
00h=Reserved
01h=Command is captured but not processed
02h=Command checking is in progress
03h=Command execution is in progress
04h-2Fh=Reserved
30h-3Fh=Custom
SUCCESS
00h=Reserved
01h=Command completed successfully
02h=Reserved
03h=Previous CMD was ABORTED by CMD Abort
04h-1Fh=Reserved
20h-2Fh=Reserved
30h-3Fh=Custom
FAILED
00h=Reserved
01h=CMDID unknown
02h=Parameter range error or parameter not supported
03h=Previous CMD was not properly ABORTED (by CMD Abort)
04h=Command checking time out
05h=CdbChkCode Error
06h=Password related error (command specific meaning)
07h=Command not compatible with operating status
</td></tr>
</table>

CdbStatus		
Bit	**Attribute**	**Description**
		08h-0Fh=Reserved for STS command checking error 10h-1Fh=Reserved 20h-2Fh=For individual STS command or task error 30h-3Fh=Custom
6	CdbHasFailed	Bool: CdbHasFailed bit indicates if there was a failure, after the module has completed execution of the last CDB command 0b: Last triggered CDB command completed successfully 1b: Last triggered CDB command failed
7	CdbIsBusy	Bool: CdbIsBusy status bit indicates whether the module is still busy, or idle and ready to accept a new CDB command 0b: Module idle, host can write 1b: Module busy, host needs to wait

12.6 Complete Remote Firmware Download (CMD 8107h to F107h)

After all blocks have been transferred, send this command to signal the completion of the remote firmware download.

- If this command fails (e.g., due to a bad image CRC), the update is considered failed, and cleanup is performed.
- If successful, proceed with the next steps

CDB Remote Complete FW Download Command Message			
Page	**Byte**	**Attribute**	**Description**
9Fh	128	CMDID	Starting from 0x8107 for Port 0 (bits [14:12] = 0), each port increments the 3-bit port index in bits [14:12], resulting in 0x9107 for Port 1, 0xA107 for Port 2, and so forth
	129		
	130	EPL length	N/A
	131		
	132	IPL length	VAR
	133	Cdb chk code	VAR
	134	RPL length	N/A
	135	RPL chk code	N/A

12.6.1 Reply Message Header

CDB Remote Complete FW Download Command (Reply)			
Page	**Byte**	**Reply**	**Description**

9Fh	134	RPL length	VAR
	135	RPL chk code	VAR

12.6.2 CDB Command Status (Page 0, byte 8 and bytes 37 – 38)

These registers are updated by the module and must be read by the host after each command to determine execution status and completion

CDB Command Status				
Page	**Byte**	**Bit**	**Attribute**	**Description**
0h	8	6	CdbCmdCompleteFlag1	Latched Flag to indicate completion of a CDB command for CDB instance 1
		7	CdbCmdCompleteFlag2	Latched Flag to indicate completion of a CDB command for CDB instance 2
	37	7 - 0	CdbStatus1	Status of the most recent CDB command in CDB instance 1
	38	7 - 0	CdbStatus2	Status of the most recent CDB command in CDB instance 2

CdbStatus		
Bit	**Attribute**	**Description**
5 – 0	CdbCommandResult	The CdbCommandResult field provides more detailed classification for each of the three coarse query results that are encoded by the pair of bit 7 (CdbIsBusy) and bit 6 (CdbHasFailed) Coarse Status / CdbIsBusy / CdbHasFailed **IN PROGRESS** / 1 / X (don't care) **SUCCESS** / 0 / 0 **FAILED** / 0 / 1 The interpretation of CdbCommandResult therefore depends on the coarse status as follows: **IN PROGRESS** 00h=Reserved 01h=Command is captured but not processed 02h=Command checking is in progress 03h=Command execution is in progress 04h-2Fh=Reserved 30h-3Fh=Custom **SUCCESS** 00h=Reserved 01h=Command completed successfully 02h=Reserved

CdbStatus		
Bit	**Attribute**	**Description**
		03h=Previous CMD was ABORTED by CMD Abort 04h-1Fh=Reserved 20h-2Fh=Reserved 30h-3Fh=Custom **FAILED** 00h=Reserved 01h=CMDID unknown 02h=Parameter range error or parameter not supported 03h=Previous CMD was not properly ABORTED (by CMD Abort) 04h=Command checking time out 05h=CdbChkCode Error 06h=Password related error (command specific meaning) 07h=Command not compatible with operating status 08h-0Fh=Reserved for STS command checking error 10h-1Fh=Reserved 20h-2Fh=For individual STS command or task error 30h-3Fh=Custom
6	CdbHasFailed	Bool: CdbHasFailed bit indicates if there was a failure, after the module has completed execution of the last CDB command 0b: Last triggered CDB command completed successfully 1b: Last triggered CDB command failed
7	CdbIsBusy	Bool: CdbIsBusy status bit indicates whether the module is still busy, or idle and ready to accept a new CDB command 0b: Module idle, host can write 1b: Module busy, host needs to wait

12.7 Run Remote Firmware Image (CMD: 8109h to F109h)

Issue this command to instruct the remote end to run the newly downloaded firmware image.

- This step may be used to switch to the new firmware version after an update.

CDB Remote Run FW Image Command Message			
Page	**Byte**	**Attribute**	**Description**
9Fh	128	CMDID	Starting from 0x8109 for Port 0 (bits [14:12] = 0), each port increments the 3-bit port index in bits [14:12], resulting in 0x9109 for Port 1, 0xA109 for Port 2, and so forth
	129		
	130	EPL length	N/A
	131		
	132	IPL length	VAR
	133	Cdb chk code	VAR

CDB Remote Run FW Image Command Message			
Page	**Byte**	**Attribute**	**Description**
	134	RPL length	N/A
	135	RPL chk code	N/A
	136		Reserved
	137	ImageToRun	00h = Traffic affecting Reset to Inactive Image. 01h = Attempt Hitless Reset to Inactive Image. 02h = Traffic affecting Reset to Running Image. 03h = Attempt Hitless Reset to Running Image
	138	DelayToReset	U16 Indicates the delay in ms after receiving this command before a reset will occur, starting from the time the CDB complete Flag is set (or NACK clearing if the CDB background mode is not set). Note: When DelayToReset is 0, the module may reset before the host has read the CdbStatus message.
	139		

12.7.1 Reply Message Header

CDB Remote Run FW Image Command (Reply)			
Page	**Byte**	**Reply**	**Description**
9Fh	134	RPL length	VAR
	135	RPL chk code	VAR

12.7.2 CDB Command Status (Page 0, byte 8 and bytes 37 – 38)

These registers are updated by the module and must be read by the host after each command to determine execution status and completion

CDB Command Status				
Page	**Byte**	**Bit**	**Attribute**	**Description**
0h	8	6	CdbCmdCompleteFlag1	Latched Flag to indicate completion of a CDB command for CDB instance 1
		7	CdbCmdCompleteFlag2	Latched Flag to indicate completion of a CDB command for CDB instance 2
	37	7 - 0	CdbStatus1	Status of the most recent CDB command in CDB instance 1
	38	7 - 0	CdbStatus2	Status of the most recent CDB command in CDB instance 2

<table>
<tr><th colspan="3">CdbStatus</th></tr>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>5 – 0</td><td>CdbCommandResult</td><td>The CdbCommandResult field provides more detailed classification for each of the three coarse query results that are encoded by the pair of bit 7 (CdbIsBusy) and bit 6 (CdbHasFailed)

<table>
<tr><td>Coarse Status</td><td>CdbIsBusy</td><td>CdbHasFailed</td></tr>
<tr><td>IN PROGRESS</td><td>1</td><td>X (don’t care)</td></tr>
<tr><td>SUCCESS</td><td>0</td><td>0</td></tr>
<tr><td>FAILED</td><td>0</td><td>1</td></tr>
</table>

The interpretation of CdbCommandResult therefore depends on the coarse status as follows:
IN PROGRESS
00h=Reserved
01h=Command is captured but not processed
02h=Command checking is in progress
03h=Command execution is in progress
04h-2Fh=Reserved
30h-3Fh=Custom
SUCCESS
00h=Reserved
01h=Command completed successfully
02h=Reserved
03h=Previous CMD was ABORTED by CMD Abort
04h-1Fh=Reserved
20h-2Fh=Reserved
30h-3Fh=Custom
FAILED
00h=Reserved
01h=CMDID unknown
02h=Parameter range error or parameter not supported
03h=Previous CMD was not properly ABORTED (by CMD Abort)
04h=Command checking time out
05h=CdbChkCode Error
06h=Password related error (command specific meaning)
07h=Command not compatible with operating status
08h-0Fh=Reserved for STS command checking error
10h-1Fh=Reserved
20h-2Fh=For individual STS command or task error
30h-3Fh=Custom</td></tr>
<tr><td>6</td><td>CdbHasFailed</td><td>Bool: CdbHasFailed bit indicates if there was a failure, after the module has completed execution of the last CDB command</td></tr>
</table>

CdbStatus		
Bit	**Attribute**	**Description**
		0b: Last triggered CDB command completed successfully 1b: Last triggered CDB command failed
7	CdbIsBusy	Bool: CdbIsBusy status bit indicates whether the module is still busy, or idle and ready to accept a new CDB command 0b: Module idle, host can write 1b: Module busy, host needs to wait

12.8 Commit Remote Firmware Image (CMD: 810Ah to F10Ah)

Once the new firmware is running and verified to be working properly, send this command to commit the running image.

- This sets the new firmware as the default image to be used after any future resets

CDB Remote Commit FW Image Command Message			
Page	**Byte**	**Attribute**	**Description**
9Fh	128	CMDID	Starting from 0x810A for Port 0 (bits [14:12] = 0), each port increments the 3-bit port index in bits [14:12], resulting in 0x910A for Port 1, 0xA10A for Port 2, and so forth
	129		
	130	EPL length	N/A
	131		
	132	IPL length	VAR
	133	Cdb chk code	VAR
	134	RPL length	N/A
	135	RPL chk code	N/A

12.8.1 Reply Message Header

CDB Remote Commit FW Image Command (Reply)			
Page	**Byte**	**Reply**	**Description**
9Fh	134	RPL length	VAR
	135	RPL chk code	VAR

12.8.2 CDB Command Status (Page 0, byte 8 and bytes 37 – 38)

These registers are updated by the module and must be read by the host after each command to determine execution status and completion

CDB Command Status				
Page	**Byte**	**Bit**	**Attribute**	**Description**
0h	8	6	CdbCmdCompleteFlag1	Latched Flag to indicate completion of a CDB command for CDB instance 1
		7	CdbCmdCompleteFlag2	Latched Flag to indicate completion of a CDB command for CDB instance 2
	37	7 - 0	CdbStatus1	Status of the most recent CDB command in CDB instance 1
	38	7 - 0	CdbStatus2	Status of the most recent CDB command in CDB instance 2

<table>
<tr><th colspan="3">CdbStatus</th></tr>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>5 – 0</td><td>CdbCommandResult</td><td>The CdbCommandResult field provides more detailed classification for each of the three coarse query results that are encoded by the pair of bit 7 (CdbIsBusy) and bit 6 (CdbHasFailed)

<table>
<tr><td>Coarse Status</td><td>CdbIsBusy</td><td>CdbHasFailed</td></tr>
<tr><td>IN PROGRESS</td><td>1</td><td>X (don’t care)</td></tr>
<tr><td>SUCCESS</td><td>0</td><td>0</td></tr>
<tr><td>FAILED</td><td>0</td><td>1</td></tr>
</table>

The interpretation of CdbCommandResult therefore depends on the coarse status as follows:
IN PROGRESS
00h=Reserved
01h=Command is captured but not processed
02h=Command checking is in progress
03h=Command execution is in progress
04h-2Fh=Reserved
30h-3Fh=Custom
SUCCESS
00h=Reserved
01h=Command completed successfully
02h=Reserved
03h=Previous CMD was ABORTED by CMD Abort
04h-1Fh=Reserved
20h-2Fh=Reserved
30h-3Fh=Custom
FAILED
00h=Reserved
01h=CMDID unknown
02h=Parameter range error or parameter not supported</td></tr>
</table>

CdbStatus		
Bit	**Attribute**	**Description**
		03h=Previous CMD was not properly ABORTED (by CMD Abort) 04h=Command checking time out 05h=CdbChkCode Error 06h=Password related error (command specific meaning) 07h=Command not compatible with operating status 08h-0Fh=Reserved for STS command checking error 10h-1Fh=Reserved 20h-2Fh=For individual STS command or task error 30h-3Fh=Custom
6	CdbHasFailed	Bool: CdbHasFailed bit indicates if there was a failure, after the module has completed execution of the last CDB command 0b: Last triggered CDB command completed successfully 1b: Last triggered CDB command failed
7	CdbIsBusy	Bool: CdbIsBusy status bit indicates whether the module is still busy, or idle and ready to accept a new CDB command 0b: Module idle, host can write 1b: Module busy, host needs to wait

13 Host Software

13.1 Telemetry SDK

An SDK defines a standard set of commands for a host to interact with a compliant transceiver. A SONiC implementation is provided. This section describes the list of commands that can be issued from the host for retrieving current link status, retrieving historical event logs, clearing event logs or counters or upgrading firmware. The intention is for the host to continuously harvest the current link status to create link performance timeline that can be streamed to a telemetry server for aggregation into a fabric-wide network map.

13.2 Transceiver Communications

All compliant transceiver communication with the local host is performed over the I2C interface. The commands issued to the local compliant module over I2C may request telemetry data from the local transceiver or from one of the remote link partners. In the case where telemetry data is requested from a remote end point, the local transceiver is responsible for retrieving the remote data and presenting it to the local host. The local host does not directly communicate with the remote link partner modules.

13.3 Command Line APIs

The following commands will be available to interact with compliant transceivers. The low level commands that are required to build these APIs are defined in this CMIS or in this document.

- Standard CMIS register access – supported from SONiC
- Remote CMIS Register Access (Ability to extract all standard CMIS data from a remote link partner) – see section 10. Note there may be multiple remote sides of the link in a breakout or shuffle application.
- Download and display all module telemetry parameters in section 8.1 for any local or remote port
- Local or Remote hitless firmware update see section 13
- R/W Epoch time see section 9.3
- Event Logs (Local) see section 9
    - Download last event – output should be human readable – not bit field
    - Download last [programmable #] of events
    - Clear event log [adjust pointers]
    - R/W event hysteresis (default 10s)
- Event Logs (Remote) see section 9
    - Download last event – output should be human readable – not bit field
    - Download last [programmable #] of events
    - Clear event log [adjust pointers]

    - R/W event hysteresis (default 10s)
- Module link rating (link health score) display – section 8.1.4
- The host switch shall record its own pre-FEC BER, uncorrectable codewords measured at its own serdes Rx input to combine with the telemetry extracted from the optical transceivers.

14 Requirements and Repository

ITEM	***REQUIREMENT***	***REFERENCE***	***MANDATORY***
1	*Complies to Special Publication Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities.*	*(SP) 800-218*	*Y*
2	*Optional: The device shall facilitate clearing the Timestamp Origin field in the Timestamp (Feature Identifier 0Eh)*	*NVMe-OPT-4*	*N*

14.1 Repository Location

To be updated once API has been uploaded - TBD.

Appendix A - Checklist for Steering Committee (SC) approval of this Specification (to be completed by contributor(s) of this Spec)

Complete all the checklist items in the table with links to the section where it is described in this spec or an external document .

Item	Status or Details	Link to detailed explanation
Is this contribution entered into the OCP Contribution Portal?	Yes or No	If no, please state the reason.
Was it approved in the OCP Contribution Portal?	Yes or No	If no, please state the reason.
Is there a Supplier(s) that is building a product based on this Spec? (Supplier must be an OCP Solution Provider)	Yes	List Supplier Name(s) Credo
Will Supplier(s) have the product available for GENERAL AVAILABILITY within 120 days?	Yes	If more time is required, please state the timeline and reason for extension request. Please have each Supplier fill out Appendix B.