CMIS Command Data Block (CDB)

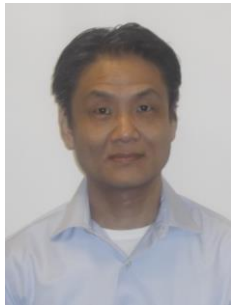
OIF Webinar

April 3rd, 2024

Hock Lim, Sr Firmware Manager; Lumentum

John Forsythe, Sr SW Systems Engineer; T.E. Connectivity

Presenters



Hock Lim,

Sr FW Manager, Lumentum

John Forsythe,

Sr SW Systems Engineer, T.E. Connectivi

Past/Future Sessions

- Overview – Gary Nichols, Ian Alderice. Dec 5th 2023
- DPSM – Doug Cattarusa, Cisco, Paul Brooks, VIAVi, February 7th 2024
- VDM – Todd Rope, Marvell – Feb 28th 2024
- CDB/FW Upgrades – Hock Lim, John Forsythe, Apr 3rd 2024
- NPSM – TBD – TBD – Volunteers.
- Others?

Webinar Overview

- **Brief History of CDB and FW Upgrades**
- Introduction to CDB
- CDB Command Summary (CMIS 5.2)
- Examples of CDB
- FW Upgrade Overview
- FW Upgrade Detail Messaging
- Future Additions to CDB

Brief History – Related to CDB

- Prior SFF standards, SFP-8472, SFF-8636, etc.
  CMIS Rev 3.0 Sept 18, 2018
  - ➢ Mgmt Interfaces are all memory mapped
    No “Standard” Firmware Upgrades

- CMIS Rev 4.0 May 08, 2019
  - ➢ CDB introduced, message based.
    Defines FW Upgrades

- CMIS Rev 4.1 June ??, 2020 – not released
  CMIS Rev 5.0 May 08, 2021
  CMIS Rev 5.1 Nov 02, 2021
  CMIS Rev 5.2 Apr 27, 2022 (Managed by OIF)
  - ➢ No major changes in CDB.
    Some definitions changed but maintains backward compatibility

- CMIS Rev 5.3 (In Straw Ballot)
  - ➢ New CDB commands extended to current set.
  - ➢ Future. CDB may be more widely used.
    (some Pros/Cons of CDB vs memory map)

Brief History

http://www.qsfp-dd.com/specification/

QSFP-DD

Home News Specification White Papers Optical Connector Co

SPECIFICATION

September 27, 2023 – QSFP-DD/QSFP-DD800/QSFP-DD1600 Hardware Specification for QSFP DOUBLE DENSITY 8X AND QSFP 4X PLUGGABLE TRANSCEIVERS – Rev 7.0

July 26, 2022 – QSFP-DD/QSFP-DD800/QSFP112 Hardware Specification for QSFP DOUBLE DENSITY 8X AND QSFP 4X PLUGGABLE TRANSCEIVERS – Rev 6.3

March 11, 2022 – QSFP-DD/QSFP-DD800/QSFP112 Hardware Specification for QSFP DOUBLE DENSITY 8X AND QSFP 4X PLUGGABLE TRANSCEIVERS – Rev 6.2

November 2, 2021 – Common Management Interface Specification (CMIS) – Rev 5.1

May 20, 2021 – QSFP-DD/QSFP-DD800/QSFP112 Hardware Specification – Rev 6.01

May 20, 2021 – Common Management Interface Specification (CMIS) – Rev 5.0

August 7, 2020 – QSFP-DD Hardware Specification for QSFP DOUBLE DENSITY 8X PLUGGABLE TRANSCEIVER – Rev 5.1

July 11, 2019 – QSFP-DD Hardware Specification for QSFP DOUBLE DENSITY 8X PLUGGABLE TRANSCEIVER – Rev 5.0

May 8, 2019 – Common Management Interface Specification – Rev 4.0

September 18, 2018 – QSFP-DD HARDWARE SPECIFICATION FOR QSFP DOUBLE DENSITY 8X PLUGGABLE TRANSCEIVER – REV 4.0

September 18, 2018 – COMMON MANAGEMENT INTERFACE SPECIFICATION FOR 8X/16X PLUGGABLE TRANSCEIVER – REV 3.0

September 19, 2017 – QSFP-DD Hardware Specification for QSFP DOUBLE DENSITY 8X PLUGGABLE TRANSCEIVER Rev 3.0

March 13, 2017 – QSFP-DD Specification for QSFP Double Density 8x Pluggable Transceiver

September 19, 2016 – QSFP-DD Specification for QSFP DOUBLE DENSITY 8X PLUGGABLE TRANSCEIVER

https://www.oiforum.com/technical-work/implementation-agreements-ias/

OIF

ABOUT OIF | TECHNICAL WORK | DOCUMENTS | MEETINGS & EVENTS | PRESS ROOM | CONTACT | MEMBER/OBSERVER LOGIN

Implementation Agreements

Management Interface Specifications

- OIF-CMIS-05.2 – Common Management Interface Specification (CMIS) *Revision 5.2* (April 2022)
  - OIF-CMIS-05.2 – Common Management Interface Specification (CMIS) *Revision 5.2* with change bars (April 2022)
- OIF-C-CMIS-01.3 – Implementation Agreement for Coherent CMIS (October 2023)
- OIF-C-CMIS-01.2 – Implementation Agreement for Coherent CMIS (March 2022)
- OIF-C-CMIS-01.1 – Implementation Agreement for Coherent CMIS (June 2020)

Webinar Overview

- Brief History of CDB and FW Upgrades
- **Introduction to CDB**
- CDB Command Summary (CMIS 5.2)
- Examples of CDB
- FW Upgrade Overview
- FW Upgrade Detail Messaging
- Future Additions to CDB

CDB Provides the following Features:

- Firmware Management and Updates
- Performance Monitoring
- Data Monitoring and Recording
- PRBS BERT Commands
- Diagnostic and Debug
- Custom CDB and Vendor Dependent CDB commands
  - Consider bringing proposal to OIF.

CMIS CDB – Command Data Block

CDB — A mechanism to support bulk data transfer into or out of the module

Use

Most common use is module upgrade where the CDB is used to pass data into the module

CDB can also be used to pull data like Performance Monitoring and/or bulk data out of the module.

Module may choose how to support CDB and advertise via Pg01h.Bytes 163-165

Method

CDB consist of:

- Page 9Fh
  - 2 bytes of unique command ID (CMDID),
  - lengths of local payload, extended payload, response.
  - Check Codes.
  - <= 120 bytes of data
  - Use for both command and response.
- Pages A0h-AFh Extended Pay Load (<=2048 bytes)
- CDB completion flags in Low Mem Byte 8.
- CDB complete status Bytes 37,38.

CMIS CDB – CDB Message Flow

Host Basic Message Flow

1. **Setup Data for CDB operation**
   - Write/s Page/s A0h-AFh (if using EPL)
2. **Trigger CDB command**
   - Write/s Page 9F
   - See P01h.B165.7 Trigger Method.
3. **Wait for CDB operation to complete**
   - NAK hand-shaking if module does not support background
   - Sleep Advertise amount of time.
   - Check CDB completion flags in Byte 8.
4. **When command completes, check Status**
   - Read Byte 37 for CDB status for Inst 0. (Inst 1 Byte 38)
   - Consist of either SUCCESS or return code or CIP (Cmd In Prog)
5. **If successful, based on CMDID**
   - Read Page 9F (Also called LPL)
   - Read Page/s A0h-Afh. (Also called EPL)
6. **Process data as needed based on CMDID**

CMD 0041 - Firmware Management Features

- Hosts need to read CMD 0041h to know what features the module supports
- Optional features like “Copy”, “Abort”, “Skip Erased Blocks”
- Duration advertisements for FW commands
- LPL/EPL support, payload size, “Start command” size
- These advertisements allow different implementations, ie saving image to RAM, erasing NVM with CMD101 or CMD107, etc

CDB Advertisements: Lower Memory – Byte 8

Byte	Bit	Name	Field Description	Type
8	7	CdbCmdCompleteFlag2	Latched Flag to indicate completion of a CDB command for CDB instance 2. Support is advertised in field 01h:163.7-6	RO/COR Adv.
	6	CdbCmdCompleteFlag1	Latched Flag to indicate completion of a CDB command for CDB instance 1. Support is advertised in field 01h:163.7-6	RO/COR Adv.
	5-3	-	Reserved	Rqd.
	2	DataPathFirmwareErrorFlag	Latched Flag to indicate that subordinated firmware in an auxiliary device for processing transmitted or received signals (e.g. a DSP) has failed.	RO/COR Adv.
	1	ModuleFirmwareErrorFlag	Latched Flag to indicate that self-supervision of the main module firmware has detected a failure in the main module firmware itself. There are several possible causes of the error such as program memory becoming corrupted and incomplete firmware loading.	RO/COR Adv.
	0	ModuleStateChangedFlag	Latched Flag to indicate a Module State Change	RO/COR Rqd.

NOTE: Beware this register is a COR register should be read by only 1 thread.

CDB Advertisements: Lower Memory Bytes 37 & 38

Byte	Bits	Register Name	Type
37	7-0	CdbStatus1	RO Adv.
38	7-0	CdbStatus2	RO Adv.

Bits	Field Name	Field 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
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
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 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

CMIS CDB – Available Options P0h.163-166.

- **Number of CDB's supported: 0, 1 or 2.**
- **CDB runs in background or foreground.**
- **CDB auto-paging supported. (benefits EPL only)**
  - **Auto-paging speeds up CDB transaction, by eliminating writes bank/page registers.**
- **Number of EPL pages supported.**
  - **These tells the host how many EPL pages are supported.**
  - **Module vendors may size this accordingly.**
  - **It may be easier for Multi-vendor environment to support all 16 pages.**
- **Max TWI transaction length 8 – 2048.**
  - **Max TWI write on memory map pages are 8 bytes.**
  - **For speed, module may advertise max sequential TWI Read or Write to be up to 2048.**
- **CDB CMDID trigger method**
  - **Trigger when 1 byte write to byte 129 in Page 9F.**
  - **Trigger when 2 byte write to bytes 128-129 in page 9F.**
  - **Trigger when any of the byte in the TWI frame consist of byte 129 in Page 9F (efficiency)**
- **CDB Max Busy Time.**
  - **How long will CDB be busy.**
  - **Two registers, one up to 80 msec, the other is 4960 msec.**

Byte	Bit	Field Name	Field Description	Type
163	7-6	CdbInstancesSupported	00b: CDB functionality not supported 01b: One CDB instance supported 10b: Two CDB instances supported 11b: Reserved **One CDB Instance** Bank 0 of Pages 9Fh and the subset of Pages A0h-AFh advertised in CdbMaxPagesEPL (01h:163.3-0), CdbCmdCompleteFlag1 Flag (00h:8.6) and the associated Mask (00h:31.6) are supported. **Two CDB Instances** Banks 0 and 1 of Pages 9Fh and of the subset of Pages A0h-AFh advertised in CdbMaxPagesEPL (01h:163.3-0), CdbCmdCompleteFlag<i> (00h:8.7-6), <i> = 1,2, and the associated Masks (00h:31.7-6) are supported.	RO Rqd.
	5	CdbBackgroundModeSupported	0b: Background CDB operation not supported. 1b: Background CDB operation supported *Note: In Background Mode, register access is possible while a CDB command is still being processed.*	RO Cnd.
	4	CdbAutoPagingSupported	When the Management Memory current address pointer advances past the end of an Extended Payload (EPL) CDB Page (A0h-AFh), the page number in the PageSelect Byte will automatically increment and the Memory Map current address pointer automatically wraps to 128. When the last page number AFh is incremented, the page number wraps back to A0h. 0b: Auto-Paging not supported	RO Cnd.
	3-0	CdbMaxPagesEPL		

Byte	Bit	Field Name	Field Description	Type
164	7-0	CdbReadWriteLengthExtension	*Note: For READ and WRITE efficiency in CDB messaging, a module can support multi-byte ACCESS in the CDB Page range (9Fh-AFh) with more than 8 bytes.* **CdbReadWriteLengthExtension = i** specifies **i***8 allowable additional number of bytes in a WRITE or READ access to an EPL CDB Page (A0Fh-AFh), i.e. **i** is a length extension in units of byte octets (8 bytes). For page **9Fh** (without auto paging support), the allowable length extension is min(**i**,15)*8 = 120 Bytes. 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 max **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) *Note: If the MCI transaction from the host is longer than the length allowed as per this advertisement, the module may ignore bytes written beyond the allowed length and*	RO Cnd.

Byte	Bit	Field Name	Field Description	Type
165	7	CdbCommandTriggerMethod	Determines how the host triggers CDB command processing in the module and when this occurs: 1b: when the MCI transaction of a WRITE access including the CMDID register 9Fh:129 is properly terminated by the host (STOP). 0b: when a single byte WRITE to 9Fh:129 or a two-byte WRITE to the CMDID register Byte 9Fh:128-129 is properly terminated by the host (STOP). *Note: Preferred method 1b enables the host to WRITE a complete CMD message (header and body) in one go, whereas in Method 0b the host composes the CMD message body first and then triggers CMD processing in a second step. See also section 7.2.*	RO Cnd.
	6-5	-	**Reserved**	RO
	4-0	CdbExtMaxBusyTime	When CdbMaxBusySpecMethod=1b: CdbExtMaxBusyTime = X encodes the maximum CDB busy time T_{CDBB} as max(1,X)*160 ms in a range of 160 ms to 4960 ms. When CdbMaxBusySpecMethod=0b: don't care	RO Cnd.
166	7	CdbMaxBusySpecMethod	0b: Indicates that the maximum CDB busy time T_{CDBB} is specified via CdbMaxBusyTime (01h:166.6-0) 1b: Indicates that the maximum CDB busy time T_{CDBB} is specified via Cdb**Ext**MaxBusyTime (01h:165.4-0).	RO Cnd.
	6-0	CdbMaxBusyTime	When CdbMaxBusySpecMethod=0b: CdbMaxBusyTime=X encodes the maximum CDB busy time T_{CDBB} as (80–max(80,X))ms in a range of 0 ms to 80 ms. When CdbMaxBusySpecMethod=1b: don't care	RO Cnd.

- **These options are provided to Module vendor to choose the best implementation based on complexity of MCU's.**

CDB Advertisements: Page 0x01 - Byte 163

Byte	Bit	Field Name	Field Description	Type
163	7-6	CdbInstancesSupported	00b: CDB functionality not supported 01b: One CDB instance supported 10b: Two CDB instances supported 11b: Reserved **One CDB Instance** Bank 0 of Pages 9Fh and the subset of Pages A0h-AFh advertised in CdbMaxPagesEPL (01h:163.3-0), CdbCmdCompleteFlag1 Flag (00h:8.6) and the associated Mask (00h:31.6) are supported. **Two CDB Instances** Banks 0 and 1 of Pages 9Fh and of the subset of Pages A0h-Afh advertised in CdbMaxPagesEPL (01h:163.3-0), CdbCmdCompleteFlag<i> (00h:8.7-6), <i> = 1,2, and the associated Masks (00h:31.7-6) are supported.	RO Rqd.
	5	CdbBackgroundModeSupported	0b: Background CDB operation not supported. 1b: Background CDB operation supported	RO Cnd.
			Note: In Background Mode, register access is possible while a CDB command is still being processed.	
	4	CdbAutoPagingSupported	When the Management Memory current address pointer advances past the end of an Extended Payload (EPL) CDB Page (A0h-Afh), the page number in the PageSelect Byte will automatically increment and the Memory Map current address pointer automatically wraps to 128. When the last page number Afh is incremented, the page number wraps back to A0h. 0b: Auto Paging not supported 1b: Auto Paging and Auto Page wrap supported	RO Cnd.
	3-0	CdbMaxPagesEPL	This field encodes the EPL Page range supported or, equivalently, the maximum length of extended payload: **Value** / **Supported EPL Pages** / **Total Number of Pages** / **EPL Bytes** 0: (none) 0 0 1: A0h 1 128 2: A0h-A1h 2 256 3 A0h-A2h 3 384 4 A0h-A3h 4 512 5 A0h-A7h 8 1024 6 A0h-Abh 12 1536 7 A0h-Afh 16 2048 *Note: A host can access all supported EPL Pages and the EPL Page range is sufficient for all supported CDB commands. The required number of EPL pages may be CDB command specific.*	RO Cnd.

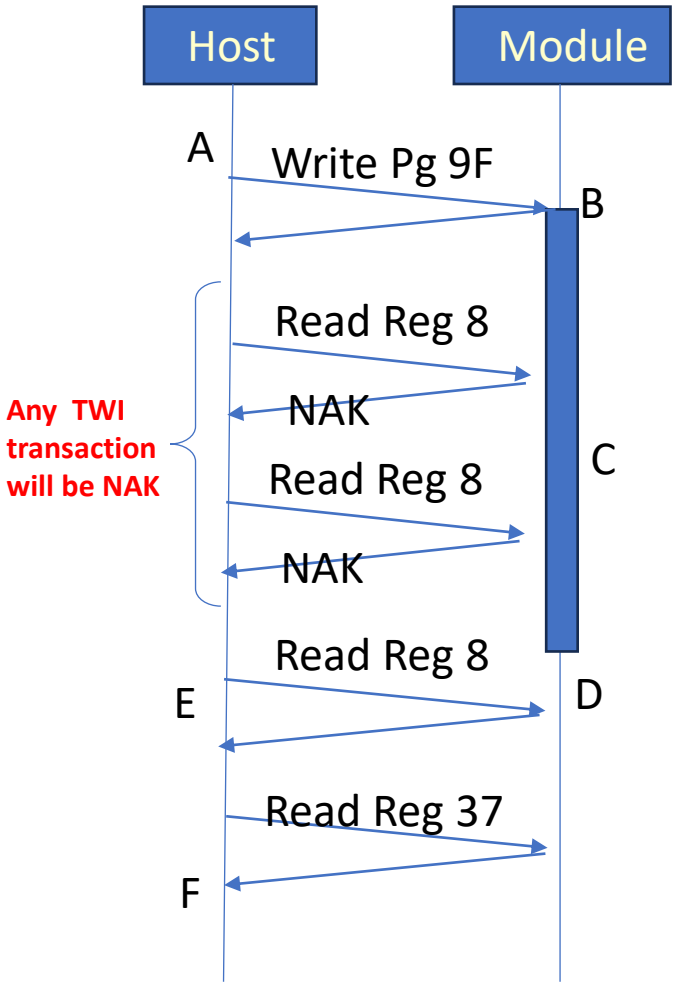
CDB Advertisements: Page 0x01 - Byte 164

Byte	Bit	Field Name	Field Description	Type
164	7-0	CdbReadWriteLengthExtension	*Note: For READ and WRITE efficiency in CDB messaging, a module can support multi-byte ACCESS in the CDB Page range (9Fh-Afh) with more than 8 bytes.* **CdbReadWriteLengthExtension** = **i** specifies **i***8 allowable additional number of bytes in a WRITE or READ access to an EPL CDB Page (A0Fh-Afh), i.e. **i** is a length extension in units of byte octets (8 bytes). For page **9Fh** (without auto paging support), the allowable length extension is min(**i**,15)*8 = 120 Bytes. 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 max **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) *Note: If the MCI transaction from the host is longer than the length allowed as per this advertisement, the module may ignore bytes written beyond the allowed length and not return more that so many bytes in a read.*	RO Cnd.

CDB Advertisements: Page 0x01 - Bytes 165, 166

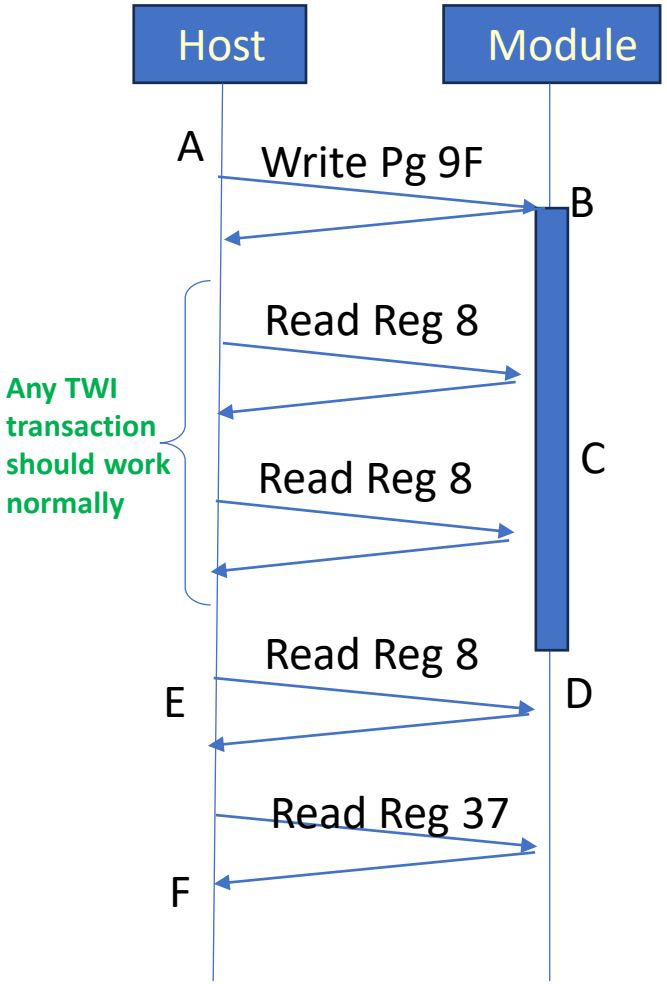
Byte	Bit	Field Name	Field Description	Type
165	7	CdbCommandTriggerMethod	Determines how the host triggers CDB command processing in the module and when this occurs: 1b: when the MCI transaction of a WRITE access including the CMDID register 9Fh:129 is properly terminated by the host (STOP). 0b: when a single byte WRITE to 9Fh:129 or a two-byte WRITE to the CMDID register Byte 9Fh:128-129 is properly terminated by the host (STOP). *Note: Preferred method 1b enables the host to WRITE a complete CMD message (header and body) in one go, whereas in Method 0b the host composes the CMD message body first and then triggers CMD processing in a second step. See also section 7.2.*	RO Cnd.
	6-5	-	**Reserved**	RO
	4-0	CdbExtMaxBusyTime	When CdbMaxBusySpecMethod=1b: CdbExtMaxBusyTime = X encodes the maximum CDB busy time $\mathbf{T}_{CDBB}$ as max(1,X)*160 ms in a range of 160 ms to 4960 ms. When CdbMaxBusySpecMethod=0b: don't care	RO Cnd.
166	7	CdbMaxBusySpecMethod	0b: Indicates that the maximum CDB busy time $\mathbf{T}_{CDBB}$ is specified via CdbMaxBusyTime (01h:166.6-0) 1b: Indicates that the maximum CDB busy time $\mathbf{T}_{CDBB}$ is specified via Cdb**Ext**MaxBusyTime (01h:165.4-0).	RO Cnd.
	6-0	CdbMaxBusyTime	When CdbMaxBusySpecMethod=0b: CdbMaxBusyTime=X encodes the maximum CDB busy time $\mathbf{T}_{CDBB}$ as (80–max(80,X))ms in a range of 0 ms to 80 ms. When CdbMaxBusySpecMethod=1b: don't care	RO Cnd.

CDB Foreground Operation. 01h:163.5=0



A. Host Setups CDB command in page 9F. Sends to Module.
B. Module RX's page 9F, executes command.
   - Starts NAK
   - Stops processing of all other TWI commands.
C. Module processing command
D. Module completes operation, set CDB completion flag Reg 8
   - CDB completion flag raised
   - Clears NAK on TWI.
E. Host may wait or continuously poll TWI
   - **TWI will NAK under Module Completes Operation**
   - **Any Other TWI will also NAK**
   - Once TWI response, command has completed.
   - CDB completion flag is also set.
F. Read Reg 37 (or 38) for command completion status.

CDB Background Operation. 01h:163.5=1

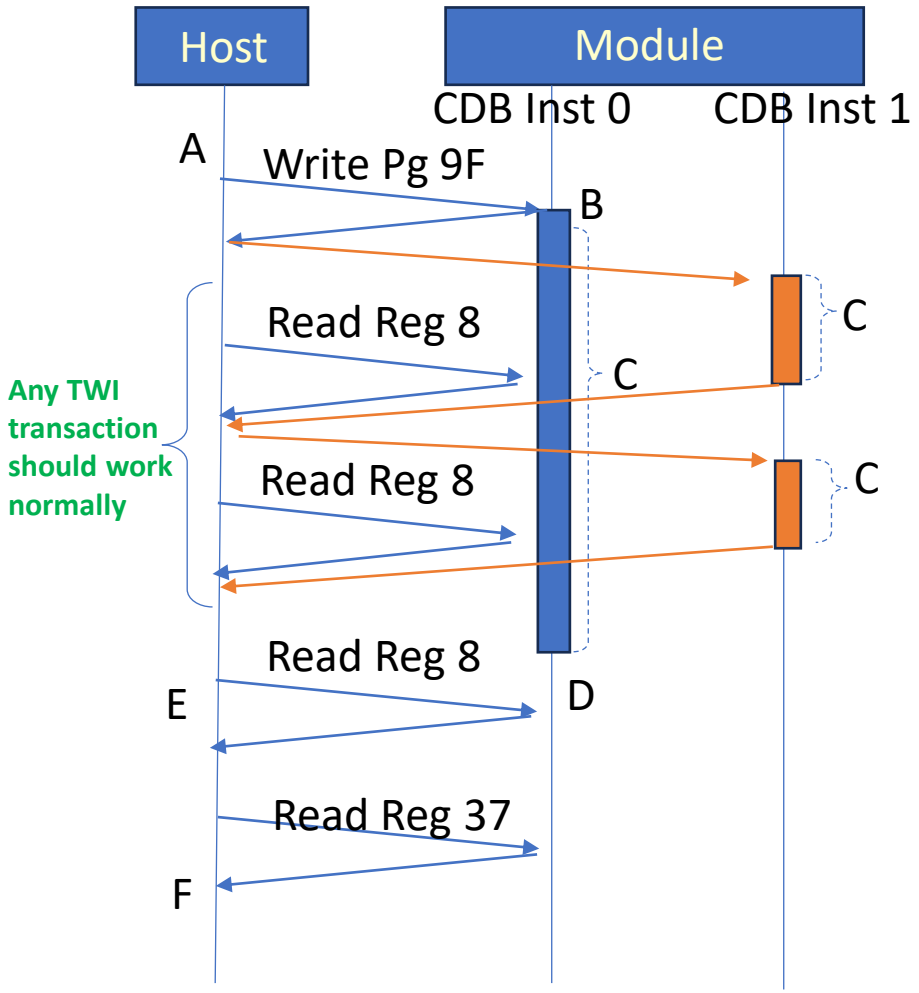


A. Host Setups CDB command in page 9F. Sends to Module.
B. Module RX's page 9F, executes command.
   - Command executes in background.
   - TWI may NAK up to 20 usec from STOP condition.
C. Module processing command
D. Module completes operation, set CDB completion flag Reg 8
   - CDB completion flag raised
   - Clears NAK on TWI.
E. Host may wait or continuously poll TWI
   - **TWI shall return any valid memory map registers**.
   - When CDB command complete, completion flag will be set in Reg 8.
   - CDB Status
F. Read Reg 37 (or 38) for command completion status.

Background CDB & 2 CDB's

- Background CDB
  - A CDB message processing command may take a longer time.
  - FW upgrade command that performs operation on flash (erase), may currently take up to 65s or more per advertisement on CMDID 0041h.
  - After CMD is sent, module performs operations in background
  - Whilst command is processing, other TWI operations, e.g. alarms & PM & VDM may run.
  - Read the CDB status when CDB to check CDB status.
- Support for Two CDB instances are optionally provided name for:
  - This is defined to make it easier for a dual threaded host software, w/o needing interlocks.
  - One host software thread uses CDB "0" for Logs, PMs & Alarms
  - One host software thread uses CDB "1" for performing Firmware Upgrades Operations
    - Start (0101h) or Write Block (0103h, 0104h) that may cause a flash write or erase.
    - These operations simply loads the new image to the "Inactive" bank. It may take a long time ~ 1 hour to load if needed.
  - Running to new downloaded image may be triggered at a later time (non peak hour)
  - NOT intended for example to use both threads simultaneous for FW upgrade.

CDB Background Operation. 01h:163.5=1



A. Host Setups CDB command in page 9F. Sends to Module.
B. Module RX's page 9F, executes command.
   - Command executes in background.
   - TWI may NAK up to 20 usec from STOP condition.
C. Module processing command
D. Module completes operation, set CDB completion flag Reg 8
   - CDB completion flag raised
   - Clears NAK on TWI.
E. Host may wait or continuously poll TWI
   - TWI shall return any valid memory map registers.
   - When CDB command complete, completion flag will be set in Reg 8.
   - CDB Status
F. Read Reg 37 (or 38) for command completion status.

NOTE: In Background Mode, Both CDB may operate at the same time.

Webinar Overview

- Brief History of CDB and FW Upgrades
- Introduction to CDB
- **CDB Command Summary (CMIS 5.2)**
- Examples of CDB
- FW Upgrade Overview
- FW Upgrade Detail Messaging
- Future Additions to CDB

Command Overview CMIS 5.2

A. Commands unchanged in CMIS Rev 4.0
B. Commands organized into groups
   - Module Commands
     - Includes Passwords
   - Capability Advertisement
   - **FW Management. (FW Upgrade)**
   - Performance Monitoring
   - Data Monitoring Recording
   - Diagnostics and Debug
   - Custom CDB commands

C. Future CMIS Rev 5.3 will define new messages
D. More features may be defined over time.

Some pros/cons of memory map interface versus CDB but we will not discuss this here.

CMD IDs		Command Group	Description	Group	See
from	to				
0000h	003Fh	Module Commands	CDB module level commands.	Rqd.	9.3
0040h	005Fh	Capability Advertisement	Query advertised CDB features and capabilities	Rqd.	9.4
0060h	006Fh	-	**Reserved**, for Bulk Read of Banks and Pages.		9.5
0070h	007Fh		**Reserved**, for Bulk Write of Banks and Pages.		9.6
0080h	00FFh	-	**Reserved**		
0100h	011Fh	Firmware Management	CDB Firmware Management	Adv.	9.7
0120h	01FFh	-	**Reserved**		
0200h	027Fh	Performance Monitoring	CDB Performance Monitoring	Adv.	9.8
0280h	02FFh	Data Recording	Non-volatile Data Recording and Monitoring	Adv.	9.9
0300h	037Fh	-	**Reserved**, for BERT functionality		9.10
0380h	03FFh	Diagnostics and Debug		Adv.	9.11
0400h	3FFFh	-	**Reserved**		
4000h	7FFFh	-	**Restricted** for use by OIF. This CMD ID range allows the OIF to define new groups of messages specific for the application. *Note: The CMD ID ranges of each group may provide additional sub-ranges restricted for use by OIF.*		
8000h	FFFFh	-	**Custom**		

Webinar Overview

- Brief History of CDB and FW Upgrades
- Introduction to CDB
- CDB Command Summary (CMIS 5.2)
- **Examples of CDB**
- FW Upgrade Overview
- FW Upgrade Detail Messaging
- Future Additions to CDB

CDB Example – Get Firmware Info

- The Host will initiate the CDB command by filling in CMD header Fields
  - Can be written in one twi transaction, must be complete header if one transaction
  - If written in a set to twi transactions, Register 129 must be in the final transaction.
  - Writing Register 129 signifies that the module can process the command
- The Module will:
  - Read the command header information
  - Process the command
  - Fill the reply data into the LPL
  - Set the CdbStatus in reg 37
  - Set the CdbCmdCompleteFlag in reg 8
- The host will:
  - Detect that the command is complete by reading register 8
  - Confirm that the command was successful in register 37
  - Read the data from the LPL

Table 9-15 CDB Command 0100h: Get Firmware Info

Page	Byte	Field Name	Description	Value
CMD Header Fields				
9Fh	128-129	CMDID	Get Firmware Info CMD ID	0100h
9Fh	130-131	EPLLength	EPL is not used	0000h
9Fh	132	LPLLength	LPL is not used	00h
9Fh	133	CdbChkCode	Check Code over 9Fh:128-132 and LPL. See Table 8-161	FEh
9Fh	134	RPLLength	*Note: Initiator may fill those reply fields, to later verify*	undef.
9Fh	135	RPLChkCode	*field updates by the target in the reply. See Table 8-161*	undef.
CMD Data (LPL)				
9Fh	136-255	-	No host-written payload	
REPLY Status				
00h	8.6 or 8.7	CdbCmdCompleteFlag	Set by module when the CDB command is complete.	1
00h	37 or 38	CdbStatus	**On Success** 00 000001b: Success **On Failure** 01 000000b: Failed, no specific failure 01 000010b: Parameter range error or not supported 01 000101b: CdbChkCode error	
REPLY Header and Data (LPL)				
9Fh	134	RPLLength	See Table 8-161	110
9Fh	135	RPLChkCode	See Table 8-161	comp.
9Fh	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* Hints: 0x00h Factory image is running (if supported) See also section 7.3.1.4 for a more detailed description.	
9Fh	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	
9Fh	138	ImageAMajor	Image A firmware major revision	
9Fh	139	ImageAMinor	Image A firmware minor revision	
9Fh	140-141	ImageABuild	Image A firmware build number	
9Fh	142-173	ImageAExtraString	Additional information	

Example – CDB CMD 0x40 “Module Features”

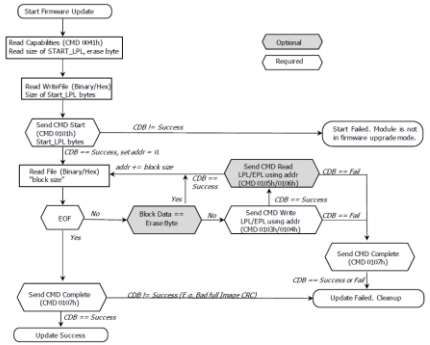
Page	Byte	Field Name	Description	Value
CMD Header Fields				
9Fh	128-129	CMDID	Module Features CMD ID	0040h
9Fh	130-131	EPLLength	EPL is not used	0000h
9Fh	132	LPLLength	LPL is not used	00h
9Fh	133	CdbChkCode	Check Code over 9Fh:128-132 and LPL. See Table 8-155	BFh
9Fh	134	RPLLength	*Note: Initiator may fill those reply fields, to later verify*	undef.
9Fh	135	RPLChkCode	*field updates by the target in the reply. See Table 8-155*	undef.
CMD Data (LPL)				
9Fh	136-255	-	**Reserved**	
REPLY Status				
00h	8.6 or 8.7	CdbCmdCompleteFlag	Set by module when the CDB command is complete.	1
00h	37 or 38	CdbStatus	**On Success** 00 000001b: Success **On Failure** 01 000000b: Failed, no specific failure 01 000101b: CdbChkCode error	
REPLY Header and Data (LPL)				
9Fh	134	RPLLength	See Table 8-155	36
9Fh	135	RPLChkCode	See Table 8-155	comp.
9Fh	136	CDB Flags	**Reserved** for additional CDB Flags.	00h
9Fh	137	-	**Reserved**	00h
9Fh	138	CMDs 0000h-0007h	Support advertisement for Module Commands A set bit indicates that the command is supported	
9Fh	139	CMDs 0008h-000Fh		
9Fh	140	CMDs 0010h-0017h		
9Fh	141	CMDs 0018h-001Fh		
9Fh	142	CMDs 0020h-0027h		
9Fh	143	CMDs 0028h-002Fh		
9Fh	144	CMDs 0030h-0037h		
9Fh	145	CMDs 0038h-003Fh		
9Fh	146	CMDs 0040h-0047h		
9Fh	147	CMDs 0048h-004Fh		
9Fh	148	CMDs 0050h-0057h		
9Fh	149	CMDs 0058h-005Fh		
9Fh	150	CMDs 0060h-0067h	Support advertisement for Bulk Read commands	
9Fh	151	CMDs 0068h-006Fh	Support advertisement for Bulk Read commands	
9Fh	152	CMDs 0070h-0077h	Support advertisement for Bulk Write commands	
9Fh	153	CMDs 0078h-007Fh	Support advertisement for Bulk Write commands	
9Fh	154-169	CMDs 0080h-00FFh		
9Fh	170-171	MaxCompletionTime	U16 Maximum CDB command execution time in ms, of all supported CDB commands. *Note: If exceeded, the host may send the CDB Abort Command.* *Note: The maximum possible MaxCompletionTime is about one minute (65.535 seconds).*	
9Fh	172-255	-	Content not specified.	

Webinar Overview

- Brief History of CDB and FW Upgrades
- Introduction to CDB
- CDB Command Summary (CMIS 5.2)
- Examples of CDB
- **FW Upgrade Overview**
- FW Upgrade Detail Messaging
- Future Additions to CDB

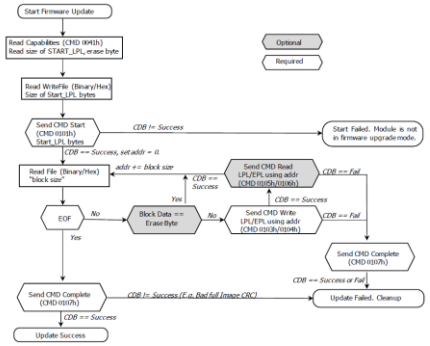
FW Management

- Provides a mechanism to upgrade firmware
  - Assumes that module has a minimum of 2 firmware images. Only InActive Image can be upgraded.
  - Assumes that module may optionally have a 3rd factory image (bootloader?)
  - Any package can be properly identified by FW Major/Minor/Build and Part Number.
  - Mechanism Supports NTA+ upgrades (as well as test commands for NTA resets)
  - Remains unchanged since CMIS Rev 4.0 (2019)
- Upgrade Process : 3 steps
  - Step 1. Downloading the New Image. (InActive Image is Replaced)
  - Step 2. Run the New Image
  - Step 3. Commit the New Image.
  - Step 1 can either be TA or NTA. This depends on module architecture.
  - Step 2 can either be TA or NTA, this requires module to support non traffic affecting resets.
  - Step 3 Commit make the running image the default image on resets. Should be NTA.
- Does not define (by design – left to vendor)
  - File Format, common header, binary, intel hex, file signatures, security, compression and encryption.
    - At time this was designed (2018) host vendors simply want to transfer the file to the module w/o processing the file.
  - Single or multiple files to upgrade various target on the Module
    - Module vendor can choose to download separate file for separate targets if they can manage the version number reporting.



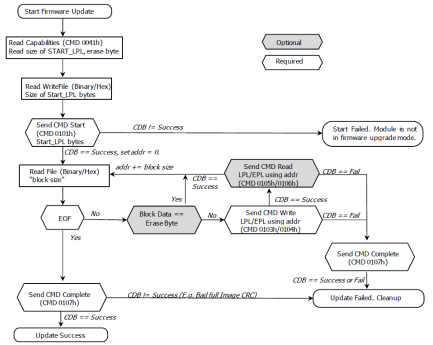
+ NTA = Non Traffic Affecting. TA = Traffic Affecting

FW Management – Multivendor 1



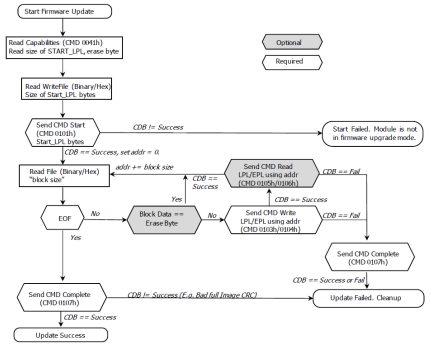
- What does “Multivendor” mean?
- Assumption is that Host performs the same action for any vendors module.
- Recommended method is to follow a standard protocol to load a single file per vendor.
  (Loading multiple files may make it difficult for host to manage various FW components within the module in a multi-vendor environment)
- Multivendor requirement is being brought up in recent OIF meetings.
  - Method already allows for this.

FW Management – Multivendor 2



- Supported by current CMIS mechanism
  - Host has to use CDB mechanism to load file into module
  - Use the recommended FW Upgrade procedure.
  - Make sure all CDB status are checked in all the messaging steps.
- Module Vendor – Module Firmware Architecture
  - Identifying or signing a file to ensure only the correct file is accepted.
  - Provide a single file to download all components.
  - Host simply download the file using the CMIS mechanism.
- Both host and module
  - Follow the recommended Upgrade Flow Diagram in CMIS.

CMIS Firmware Upgrade Flow



Step 1
Download to new Image

Read to Confirm capabilities. EPL support is key value to check for.

Starts the upgrade process. Data includes file size of upgrade file.

Send blocks using command 103h (LPL) or 104h (EPL) until the complete file has been sent.

Once file has been completely sent, send command 107h. This will validate the transferred file and copy the transferred file into inactive image

Step 2
Run to new Image

Once file has been validated, send command 109h and specify run from inactive image (hitless/not). This will boot the module out of Bank B.

Step 3
Commit the Running Image

Once module is running, send command 10Ah to commit the load and copy the load to Bank A.

Optionally, send command 108h to copy the load to the other bank.

Read Capabilities
CMD 041h

Start FW download
CMD 101h

Read File

EOF

Send Block
CMD 103h/104h

Complete FW download
CMD 107h

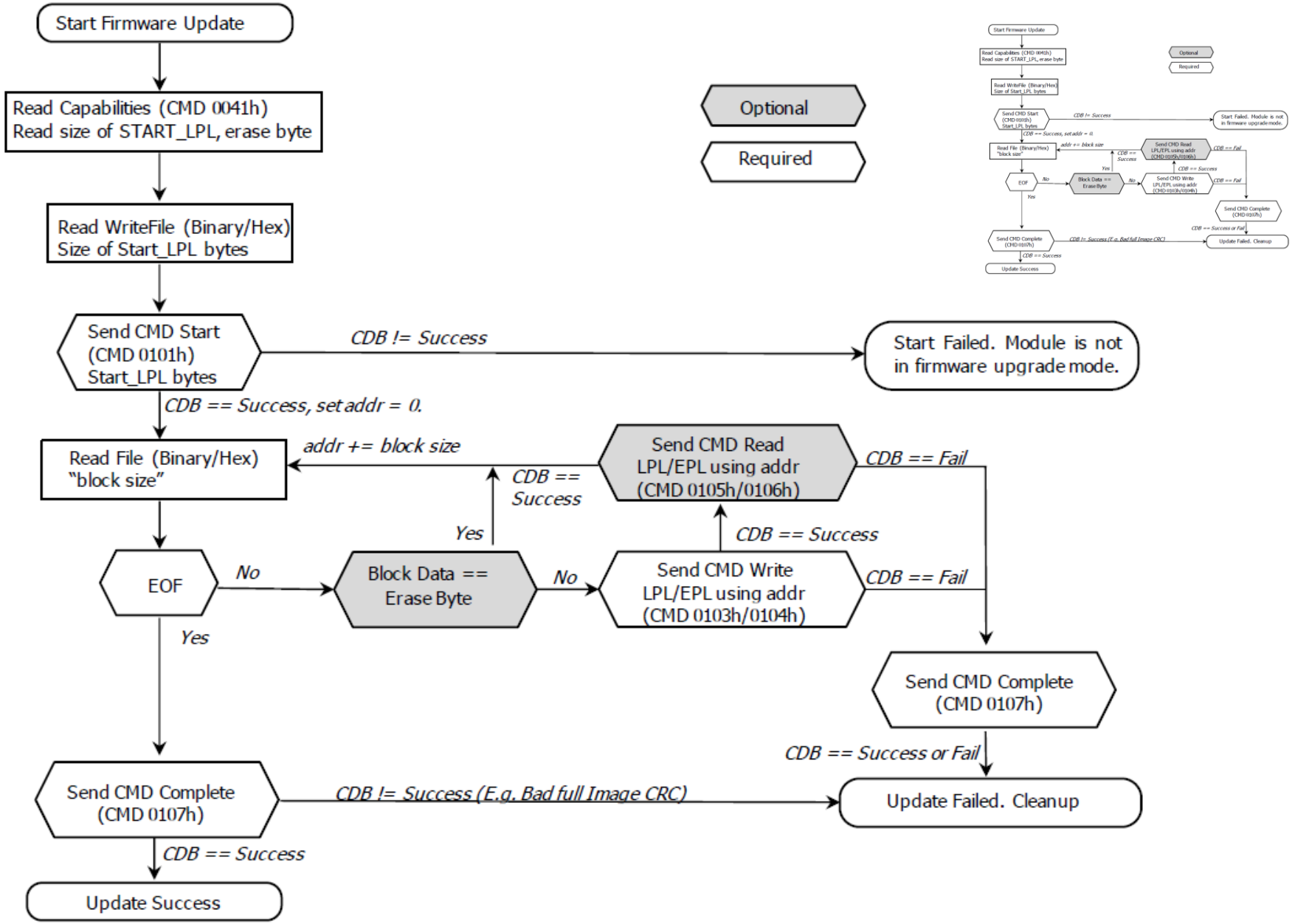
Run FW image
CMD 109h

Commit FW image
CMD 10Ah

Copy FW image
CMD 108h

If "copy" command supported.
Otherwise repeat Step 1 above.

Firmware Update Flow Diagram



Webinar Overview

- Brief History of CDB and FW Upgrades
- Introduction to CDB
- CDB Command Summary (CMIS 5.2)
- Examples of CDB
- **FW Upgrade Overview**
- **FW Upgrade Detail Messaging**
- Future Additions to CDB

FW Upgrade Messages

CDB Upgrade CMDID

1. 0x0100: Get FW Info. Running, Valid, Committed.
2. 0x0101: Start Firmware Download.
3. 0x0102: Abort
4. 0x0103: Write Data using LPL
5. 0x0104: Write Data using EPL
6. 0x0105: Read Data using LPL
7. 0x0106: Read Data using EPL
8. 0x0107: Complete
9. 0x0108: Copy
10. 0x0109: Run
11. 0x010A: Commit

NOTE: Command Execution Time for Firmware Management messages may take longer than 4960 msec and are separately advertised in CMDID 0041h.

ID	Command Title	Description
0100h	Get Firmware Info	When the host issues this command, the module returns the requested FW information of all field-updateable firmware in the module.
0101h	Start Firmware Download	The host issues this command to initiate a firmware update. The module may completely erase the target or simply acknowledge and prepare the module firmware for any appropriate update or dynamically erase as each data block arrives. On success, the host may begin sending the firmware image using command codes 0103h-0107h.
0102h	Abort Firmware Download	Aborts the FW Download if a FW Start has been issued.
0103h	Write Firmware Block LPL	With this command, the host downloads a firmware image block previously stored in the LPL area. The module transfers that firmware image block into non-volatile storage. *Note: Each image block transfer from host to module is covered by the CDB command block checksum; this checksum does not ensure that the image has been properly transferred to non-volatile storage.*
0104h	Write Firmware Block EPL	With this command, the host downloads a firmware image block previously stored in the EPL Page(s). The module transfers that firmware image block into non-volatile storage.
0105h	Read Firmware Block LPL	The host may use this command to read back the firmware image that was most recently written to non-volatile storage. The module copies the image from non-volatile storage to the LPL Page. The firmware image transfer from the module to the host is covered by the CDB command block checksum, but this checksum does not ensure that the image has been properly transferred from non-volatile storage to the LPL Page.
0106h	Read Firmware Block EPL	The host may use this command to read back the firmware image that was most recently written to non-volatile storage. The module copies the image from non-volatile storage to the EPL Page(s). The firmware image transfer from the module to host is covered by a block checksum, but this checksum does not ensure that the image has been properly transferred from non-volatile storage to the EPL Page(s).
0107h	Complete Firmware Download	The host issues this command when the entire firmware image has been written via LPL or via EPL Pages, or to stop the download after failure. If this command is not issued, the firmware cannot be run even if the image is properly loaded to non-volatile storage. The module validates the checksum associated with the image when the host issues this command.

ID	Command Title	Description
0108h	Copy Firmware Image	When multiple images are supported for a given subsystem in the module, this command causes the module to copy an image from one non-volatile storage location to another one. It is assumed that the copy firmware image command includes a validation process by the module firmware to ensure the copied image is valid. The CDB complete firmware image command 0107h does not need to be called after a copy firmware image command.
0109h	Run Firmware Image	This command is used to start and run a selected image. This command transfers control from the currently running firmware to a selected firmware that is started. It can be used to switch between firmware versions, or to perform a restart of the currently running firmware.
010Ah	Commit Firmware Image	The host uses this command to commit the running image so that the module will boot from it on future boots. *The assumption is that the host has a time period where it runs the new firmware and "accepts" the new firmware. During this time, if a reset occurs, the previously committed image will be run. When the host issues this command, the module will mark a non-volatile storage location to be used after future module resets. Only the running image can be committed. This is to avoid committing a "bad" image.*
010Bh - 011Fh	-	**Reserved**

CMDID 0100h: Firmware Get Information

Information includes:

- Firmware Status: Which of Bank A or B is running (Active)
- Image Versions
- Image Extra String is for vendor to put any additional information about the package.
  - Information to display not specified
  - Recommended to be ASCII
  - show date, time stamps
  - show FW version of other components
  - Any information you want the host to be able to return for detail FW identification.

REPLY Header and Data (LPL)				
9Fh	134	RPLLength	See Table 8-155	110
9Fh	135	RPLChkCode	See Table 8-155	comp.
9Fh	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* Hints: 0x00h Factory image is running (if supported) See also section 7.3.1.4 for a more detailed description.	
9Fh	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	
9Fh	138	ImageAMajor	Image A firmware major revision	
9Fh	139	ImageAMinor	Image A firmware minor revision	
9Fh	140-141	ImageABuild	Image A firmware build number	
9Fh	142-173	ImageAExtraString	Additional information	
9Fh	174	ImageBMajor	Image B firmware major revision	
9Fh	175	ImageBMinor	Image B firmware minor revision	
9Fh	176-177	ImageBBuild	Image B firmware build number	
9Fh	178-209	ImageBExtraString	Additional information	
9Fh	210	FactoryBootMajor	Factory or Boot image firmware major revision	
9Fh	211	FactoryBootMinor	Factory or Boot image firmware minor revision	
9Fh	212-213	FactoryBootBuild	Factory or Boot image firmware build number	
9Fh	214-245	FactoryBootExtraStr	Additional information	
9Fh	246-255	-	Content unspecified	

CMDID 0101h: Start Firmware Download

- Tells the module that Firmware Download is about to start
- Firmware in InActive Bank (A or B) will be upgraded
- Firmware in InActive Bank (A or B) will be erased.
- Image size is conveyed. This is typically file size.
- Include up to 112 bytes of Optional Vendor Data
  - Use Not recommended for multi vendor InterOp
  - Difficult to convey to host optional data size in Start CMDID
  - Vendor can still use these optional bytes for additional flags, instructions, configurations for special download modes.
- Subsequent Write Block (0x0103h or 0x0104h) may not work without a Start Command.

Table 9-16 CDB Command 0101h: Start Firmware Download

Page	Byte	Field Name	Description	Value
CMD Header Fields				
9Fh	128-129	CMDID	Start firmware download process CMD ID	0101h
9Fh	130-131	EPLLength	EPL is not used	0000h
9Fh	132	LPLLength	LPL length	comp.
9Fh	133	CdbChkCode	Check Code over 9Fh:128-132 and LPL. See Table 8-161	comp.
9Fh	134	RPLLength	*Note: Initiator may fill those reply fields, to later verify field updates by the target in the reply. See Table 8-161*	undef.
9Fh	135	RPLChkCode		undef.
CMD Data (LPL)				
9Fh	136-139	ImageSize	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.	var.
9Fh	140-143	-	**Reserved**	0h
9Fh	144-255	VendorData	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.	
REPLY Status				
00h	8.6 or 8.7	CdbCmdCompleteFlag	Set by module when the CDB command is complete.	1
00h	37 or 38	CdbStatus	**In Progress** 10 000001b: Busy processing command, CMD captured 10 000010b: Busy processing command, CMD checking 10 000011b: Busy processing command, CMD execution **On Success** 00 000001b: Success **On Failure** 01 000000b: Failed, no specific failure 01 000010b: Parameter range error or not supported 01 000101b: CdbChkCode error	
REPLY Header and Data (LPL)				
9Fh	134	RPLLength	See Table 8-161	0
9Fh	135	RPLChkCode	See Table 8-161	0
9Fh	136-255	-	No data returned. Content not specified.	

CMDID 0102h: Abort Firmware Download

- Abort any ongoing Firmware Download in progress on InActive.
  - Could happen if Host Resets in during Firmware Upgrades.
- Should have no ill effect if no Firmware Upgrade is in progress.
- InActive Image may be erased, invalid or untouched.
  (depends on module vendors capabilities).
- Use CMDID 0100h to read FW Status after the Abort command.
- May need to restore and load valid image in InActive Bank, especially if operator desires to have the inactive image as a backup image.

Table 9-17 CDB Command 0102h: Abort Firmware Download

Page	Byte	Field Name	Description	Value
CMD Header Fields				
9Fh	128-129	CMDID	Abort Firmware Download CMD ID	0102h
9Fh	130-131	EPLLength	EPL is not used	0000h
9Fh	132	LPLLength	LPL is not used	00h
9Fh	133	CdbChkCode	Check Code over 9Fh:128-132 and LPL. See Table 8-161	FCh
9Fh	134	RPLLength	*Note: Initiator may fill those reply fields, to later verify*	undef.
9Fh	135	RPLChkCode	*field updates by the target in the reply. See Table 8-161*	undef.
CMD Data (LPL)				
9Fh	136-255	No host-written payload		
REPLY Status				
00h	8.6 or 8.7	CdbCmdCompleteFlag	Set by module when the CDB command is complete.	1
00h	37 or 38	CdbStatus	**In Progress** 10 000001b: Busy processing command, CMD captured 10 000010b: Busy processing command, CMD checking 10 000011b: Busy processing command, CMD execution **On Success** 00 000001b: Success **On Failure** 01 000000b: Failed, no specific failure 01 000010b: Parameter range error or not supported 01 000101b: CdbChkCode error	
REPLY Header and Data (LPL)				
9Fh	134	RPLLength	See Table 8-161	0
9Fh	135	RPLChkCode	See Table 8-161	0
9Fh	136-255	-	No data returned. Content not specified.	

CMDID 0103h/0104h: Write Data Block

- CMDID 0103h. Can write up to 116 bytes per TWI transaction.
- CMDID 0104h. Write Data Block through EPL.
  - AutoPaging Support
  - Write block size (TWI write size) needs to be per Advertised.
  - P01h.B163 – number of EPL pages
  - P01h.B164 – max number of bytes per TWI transaction.
  - Multi Vendor
    - Read module capabilities.
    - Transfer Data accordingly.
- A Write Operation will transfer a block of data to the module from a file.
- A Write Operation may take different execution times
  - FW Signature of the file to be validated (ensuring file can be accepted)
  - Flash Erasure
  - Flash Writing Operation
  - Internal Write to temporary memory of any kind.
  - Best for Host to use completion flags and status to manage transfer
  - Max Write Execution Time is defined in CMDID 0041h.

Table 9-18 CDB Command 0103h: Write Firmware Block LPL

Page	Byte	Field Name	Description	Value
CMD Header Fields				
9Fh	128-129	CMDID	Write Firmware Block LPL CMD ID	0103h
9Fh	130-131	EPLLength	EPL is not used	0000h
9Fh	132	LPLLength	The actual length of the firmware block in the FirmwareBlock field + 4.	comp.
9Fh	133	CdbChkCode	Check Code over 9Fh:128-132 and LPL. See Table 8-161	comp.
9Fh	134	RPLLength	*Note: Initiator may fill those reply fields, to later verify field updates by the target in the reply. See Table 8-161*	undef.
9Fh	135	RPLChkCode		undef.
CMD Data (LPL)				
9Fh	136-139	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.	
9Fh	140-255	FirmwareBlock	One block of the firmware image. The actually needed length may be shorter than the available FirmwareBlock field size. This actual length of the block is defined in Byte 132 (LPLLength), see above.	
REPLY Status				
00h	8.6 or 8.7	CdbCmdCompleteFlag	Set by module when the CDB command is complete.	1
00h	37 or 38	CdbStatus	**In Progress** 10 000001b: Busy processing command, CMD captured 10 000010b: Busy processing command, CMD checking 10 000011b: Busy processing command, CMD execution **On Success** 00 000001b: Success **On Failure** 01 000000b: Failed, no specific failure 01 000010b: Parameter range error or not supported 01 000101b: CdbChkCode error	
REPLY Header and Data (LPL)				
9Fh	134	RPLLength	See Table 8-161	0
9Fh	135	RPLChkCode	See Table 8-161	0
9Fh	136-255	-	No data returned. Content not specified.	

CMDID 0105h/0106h: Read Data Block

- Read Block Operations may not be supported
  - Modules/Vendors may not want Readback for Security.
- If Read is Supported
  - Data should be read back from NVM and processed accordingly. (Not from a RAM copy.)
  - Multi Vendor
    - data read needs to be compared to binary file.
    - Filesize to be used and number of bytes to be read.
- Read Block Through LPL
  - Max bytes is 116.
- Read Block Through EPL
  - TWI Read Size needs to be based on module advertisement
  - Auto Paging, TWI read max size etc

Table 9-20 CDB Command 0105h: Read Firmware Block LPL

Page	Byte	Field Name	Description	Value
CMD Header Fields				
9Fh	128-129	CMDID	Read Firmware Block LPL CMD ID	0105h
9Fh	130-131	EPLLength	EPL is not used	0000h
9Fh	132	LPLLength	LPL length	06h
9Fh	133	CdbChkCode	Check Code over 9Fh:128-132 and LPL. See Table 8-161	comp.
9Fh	134	RPLLength	*Note: Initiator may fill those reply fields, to later verify field updates by the target in the reply. See Table 8-161*	undef.
9Fh	135	RPLChkCode		undef.
CMD Data (LPL)				
9Fh	136-139	BlockAddress	U32 Starting byte address of this block of data within the supplied image file minus the size of the size of the "Start Command Payload Size". See section 7.3.1.2.	
9Fh	140-141	Length	U16 Number of bytes to read back to the LPL in this command, starting at the indicated address.	
9Fh	142-255	-	**Reserved**	
REPLY Status				
00h	8.6 or 8.7	CdbCmdCompleteFlag	Set by module when the CDB command is complete.	1
00h	37 or 38	CdbStatus	**In Progress** 10 000001b: Busy processing command, CMD captured 10 000010b: Busy processing command, CMD checking 10 000011b: Busy processing command, CMD execution **On Success** 00 000001b: Success **On Failure** 01 000000b: Failed, no specific failure 01 000010b: Parameter range error or not supported 01 000101b: CdbChkCode error	
REPLY Header and Data (LPL)				
9Fh	134	RPLLength	See Table 8-161	var.
9Fh	135	RPLChkCode	See Table 8-161	comp.
9Fh	136-139	Address of block	U32 Base address of the data block within the firmware image	
9Fh	140-255	ImageData	Up to 116 Bytes	

CMDID 0107h: Complete

- Complete tells the module that all the data has been transmitted.
- Module may validate the entire downloaded image.
- MultiVendor NOTE:
  - Some module may not erase the InActive Image until image is validated.
  - Some module may already erase the InActive Image.
- If Complete fails
  - Use CMDID 0100h to Read the Firmware Status.

Table 9-22 CDB Command 0107h: Complete Firmware Download

Page	Byte	Field Name	Description	Value
CMD Header Fields				
9Fh	128-129	CMDID	Complete Firmware Download CMD ID	0107h
9Fh	130-131	EPLLength	EPL is not used	0000h
9Fh	132	LPLLength	LPL is not used	00h
9Fh	133	CdbChkCode	Check Code over 9Fh:128-132 and LPL. See Table 8-161	F7h
9Fh	134	RPLLength	*Note: Initiator may fill those reply fields, to later verify field updates by the target in the reply. See Table 8-161*	undef.
9Fh	135	RPLChkCode		undef.
CMD Data (LPL)				
9Fh	136-255	-	No host-written payload	
REPLY Status				
00h	8.6 or 8.7	CdbCmdCompleteFlag	Set by module when the CDB command is complete.	1
00h	37 or 38	CdbStatus	**In Progress** 10 000001b: Busy processing command, CMD captured 10 000010b: Busy processing command, CMD checking 10 000011b: Busy processing command, CMD execution **On Success** 00 000001b: Success **On Failure** 01 000000b: Failed, no specific failure 01 000010b: Parameter range error or not supported 01 000101b: CdbChkCode error	
REPLY Header and Data (LPL)				
9Fh	134	RPLLength	See Table 8-161	0
9Fh	135	RPLChkCode	See Table 8-161	0
9Fh	136-255	-	Nothing returned	

CMDID 0108h: Copy

- Purpose
  - Allow both images to be the same running version as defined by vendor
  - Minimize host interaction
- Copy command may not be supported by the module
  - Modules running directly out of Flash.
  - Same process (Start, Write Block, Complete) need to be use to update InActive image.
- Copy Command if supported
  - Shall not affect normal module operation during the copy process.
  - Shall be non traffic affecting.
  - Shall validate the image after the copy process

Table 9-23 CDB Command 0108h: Copy Firmware Image

Page	Byte	Field Name	Description	Value
CMD Header Fields				
9Fh	128-129	CMDID	Copy Firmware Image CMD ID	0108h
9Fh	130-131	EPLLength	EPL is not used	0000h
9Fh	132	LPLLength	LPL length	01h
9Fh	133	CdbChkCode	Check Code over 9Fh:128-132 and LPL. See Table 8-161	comp.
9Fh	134	RPLLength	*Note: Initiator may fill those reply fields, to later verify*	undef.
9Fh	135	RPLChkCode	*field updates by the target in the reply. See Table 8-161*	undef.
CMD Data (LPL)				
9Fh	136	Copy Direction	ABh: Copy Image A into Image B BAh: Copy Image B into Image A	
9Fh	137-255	-	**Reserved**	
REPLY Status				
00h	8.6 or 8.7	CdbCmdCompleteFlag	Set by module when the CDB command is complete.	1
00h	37 or 38	CdbStatus	**In Progress** 10 000001b: Busy processing command, CMD captured 10 000010b: Busy processing command, CMD checking 10 000011b: Busy processing command, CMD execution **On Success** 00 000001b: Success **On Failure** 01 000000b: Failed, no specific failure 01 000010b: Parameter range error or not supported 01 000101b: CdbChkCode error	
REPLY Header and Data (LPL)				
9Fh	134	RPLLength	See Table 8-161	6
9Fh	135	RPLChkCode	See Table 8-161	comp.
9Fh	136-139	Length	U32 Number of bytes copied	
9Fh	140	CopyDirection	ABh: Image A was copied into Image B BAh: Image B was copied into Image A	
9Fh	141	CopyStatus	00h : Copy Successful 01h : Copy Failed	
9Fh	142-255	-	**Reserved**	

CMDID 0109h: Run Image

- Run the InActive image or Reset the running Active Image. Options
  - TA Traffic Affecting
  - NTA Non Traffic Affecting
  - InActive Image does not have to be newly download
- NTA are best effort
  - Module Dependent
  - Hardware Dependent
  - Some control loops may freeze for some time.
  - Require module to reach steady state operation (temperature)
- Running to New Image
  - After running to new image, do not “Commit” immediately.
  - Wait for traffic and alarms to be clear.
  - Although image may be valid, it may instability with hardware.
  - Validate the functionality (e.g. Traffic running, no alarms) before running Commit (next slide).

Table 9-24 CDB Command 0109h: Run Firmware Image

Page	Byte	Field Name	Description	Value
CMD Header Fields				
9Fh	128-129	CMDID	Run FW Image CMD ID	0109h
9Fh	130-131	EPLLength	EPL is not used	0000h
9Fh	132	LPLLength	LPL length	04h
9Fh	133	CdbChkCode	Check Code over 9Fh:128-132 and LPL. See Table 8-161	comp.
9Fh	134	RPLLength	*Note: Initiator may fill those reply fields, to later verify*	undef.
9Fh	135	RPLChkCode	*field updates by the target in the reply. See Table 8-161*	undef.
CMD Data (LPL)				
9Fh	136	-	**Reserved**	
9Fh	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	
9Fh	138-139	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.*	
9Fh	140-255	-	**Reserved**	
REPLY Status				
00h	8.6 or 8.7	CdbCmdCompleteFlag	Set by module when the CDB command is complete.	1
00h	37 or 38	CdbStatus	**In Progress** 10 000001b: Busy processing command, CMD captured 10 000010b: Busy processing command, CMD checking 10 000011b: Busy processing command, CMD execution **On Success** 00 000001b: Success **On Failure** 01 000000b: Failed, no specific failure 01 000010b: Parameter range error or not supported 01 000101b: CdbChkCode error	
REPLY Header and Data (LPL)				
9Fh	134	RPLLength	See Table 8-161	
9Fh	135	RPLChkCode	See Table 8-161	
9Fh	136-255	-	Nothing returned. Contents unspecified	

CMDID 010Ah: Commit Image

- Makes the current Image the Default Image
  - Hot Plug (Cold Starts)
  - Software or Watch Dog resets.
- A module shall NOT commit the InActive Image.
  - Prevent InActive Image from being committed.
  - An InActive image may not be a “sane load”.
- Flash Operation or EEPROM operation
  - May erase flash or eeprom or any NVM indicating default image.
  - May write flash

Table 9-25 CDB Command 010Ah: Commit Image

Page	Byte	Field Name	Description	Value
CMD Header Fields				
9Fh	128-129	CMDID	Commit Image CMD ID	010Ah
9Fh	130-131	EPLLength	EPL is not used	0000h
9Fh	132	LPLLength	LPL is not used	00h
9Fh	133	CdbChkCode	Check Code over 9Fh:128-132 and LPL. See Table 8-161	comp.
9Fh	134	RPLLength	*Note: Initiator may fill those reply fields, to later verify field updates by the target in the reply. See Table 8-161*	undef.
9Fh	135	RPLChkCode		undef.
CMD Data (LPL)				
9Fh	136-255	-	**Reserved**	
REPLY Status				
00h	8.6 or 8.7	CdbCmdCompleteFlag	Set by module when the CDB command is complete.	1
00h	37 or 38	CdbStatus	**In Progress** 10 000001b: Busy processing command, CMD captured 10 000010b: Busy processing command, CMD checking 10 000011b: Busy processing command, CMD execution **On Success** 00 000001b: Success **On Failure** 01 000000b: Failed, no specific failure 01 000010b: Parameter range error or not supported 01 000101b: CdbChkCode error	
REPLY Data (LPL)				
9Fh	134	RPLLength	See Table 8-161	0
9Fh	135	RPLChkCode	See Table 8-161	0
9Fh	136-255	-	Nothing returned. Contents unspecified	

Webinar Overview

- Brief History of CDB and FW Upgrades
- Introduction to CDB
- CDB Command Summary (CMIS 5.2)
- Examples of CDB
- **FW Upgrade Overview**
- FW Upgrade Detail Messaging
- **Future additions to CDB**

What's Coming Next - CMIS 5.3

No backwards incompatibility issues

- CMD 0044h: "Security Features and Capabilities"
- CMD 0045h: "Externally Defined Features"
- CMD 0050h: "Get Application Attributes" (CMIS 5.3 supports up to 240 Application Descriptors)
- CMD 0051h: "Get Interface Code Description"
- CMD 0220h: "Get Data Path RMON Statistics"
- CMD 0230h: "Control FEC Symbol Error Weight Histogram"
- CMD 0231h: "Get FEC Symbol Error Weight Histogram"
- CMD 0232h: "Control Max FEC Symbol Error Weight"
- CMD 0233h: "Get Max FEC Symbol Error Weight"
- CMD 0400h: "Get Initial Device ID Certificate"
- CMD 0401h: "Set Signature Digest"
- CMD 0402h: "Get Digest Signature"

Q & A

Thank You