

# ENDL TEXAS

Date: 18 May 2007  
 To: T10 Technical Committee  
 From: Ralph O. Weber  
 Subject: SPC-4: Extended SCSI Commands

## Revision History

- r1 Incorporated changes requested by January '07 CAP working group.
- r2 Totally rewrote based on discussions in March '07 CAP working group, concepts summarized in 07-158r0, and discussions in April '07 CAP/Security working group (minutes in 07-182).
- r3 Incorporated minor changes requested by the May '07 CAP working group (minutes in 07-212).

Changes between r2 and r3 are indicated by change bars.

## History

In IPsec and FC-SP, transport layer security encapsulates frames in a technology called Encapsulated Security Protocol (ESP). The encapsulation adds security related information to frames that may be otherwise unchanged. An underlying assumption of ESP encapsulation is that the encapsulation data does not need to reference the encapsulated data.

Numerous attempts to add additional data to all SCSI commands (e.g., classification and priority) led to attempts to model an ESP-like encapsulation for SCSI CDBs in revisions 0 and 1 of this proposal. These attempts failed at every turn because the people with plans to add additional data wanted knowledge of the 'encapsulated' CDB's contents to facilitate their command-processing designs.

Starting with r2, this proposal describes a method for appending additional bytes of typed information to CDBs that are defined elsewhere in the SCSI command standards, including but not limited to SPC-4.

## Proposed SPC-4 Changes

Reference SPC-4 r08. Blue for new text. ~~Red-strikeout~~ for removed text. Green for proposer's notes.

{{New Definition}}

**3.1.x Extended CDB (XCDB):** A CDB that contains another CDB and additional information to support extra processing (e.g., security features). See 4.3.4.

{{New Acronym}}

**XCDB** Extended CDB (see 3.1.x)

## 4.3 The Command Descriptor Block (CDB)

### 4.3.1 CDB usage and structure

A command is communicated by sending a command descriptor block (CDB) to the device server. For several commands, the CDB is accompanied by a list of parameters in the Data-Out Buffer. See the specific commands for detailed information.

If a logical unit validates reserved CDB fields and receives a reserved field within the CDB that is not zero, then the logical unit shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

If a logical unit receives a reserved CDB code value in a field other than the OPERATION CODE field, then the logical unit shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

The fixed length CDB formats are described in 4.3.2. The variable length CDB formats are described in 4.3.3. [The XCDB format is described in 4.3.4](#). The CDB fields that are common to most commands are described in 4.3.5. The fields shown in 4.3.2 and 4.3.3 and described in 4.3.5 are used consistently by most commands. However, the actual usage of any field (except [the OPERATION CODE field](#) and [the CONTROL field](#)) is described in the subclause defining that command. If a device server receives a CDB containing an operation code that is invalid or not supported, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID COMMAND OPERATION CODE.

For all commands, if there is an invalid parameter in the CDB, the device server shall terminate the command without altering the medium.

[The XCDB format \(see 4.3.5\) is a CDB in which additional information is appended to another CDB. The requirements in this subclause apply to the XCDB and the CDB contained in the XCDB.](#)

### 4.3.2 The fixed length CDB formats

...

### 4.3.3 The variable length CDB formats

...

### 4.3.4 Extended CDBs

#### 4.3.4.1 XCDB model

[With one exception, any SCSI CDB may be extended in an XCDB. An XCDB shall not be extended in another XCDB. Instead, additional XCDB descriptors shall be added to the existing XCDB.](#)

[XCDB descriptors may be:](#)

- [a\) Added by application clients and removed by device servers, or](#)
- [b\) Added and removed by entities outside the scope of this standard that transport or process CDBs and XCDBs during their transfer from an application client to a device server.](#)

**4.3.4.2 The XCDB format**

The first byte of an XCDB shall contain the operation code 7Eh. The CONTROL byte is the CONTROL byte in the CDB field (see table x1).

**Table x1 — ESC CDB format**

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (7Eh)							
1	Reserved							
2	LENGTH (n-3)							
3	CDB							
4	CDB							
i	CDB							
XCDB descriptors								
k	First XCDB descriptor							
	:							
	:							
	Last XCDB descriptor							
n	Last XCDB descriptor							

The LENGTH field contains the number of bytes that follow in the XCDB.

The CDB field contains the CDB to which XCDB descriptors are being appended.

*{{INVALID XCDB is a new additional sense code.}}*

Each XCDB descriptor (see table x2) contains fields associated with a specified extension type (see table x3). The command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID XCDB, if:

- a) An XCDB does not contain at least one XCDB descriptor;
- b) More than one XCDB descriptor contains the same extension type; or
- c) The order of extension types in the XCDB descriptors differs from that shown in table x3.

**Table x2 — XCDB descriptor format**

Bit Byte	7	6	5	4	3	2	1	0
0	EXTENSION TYPE							
1	Extension parameters							
n	Extension parameters							

The EXTENSION TYPE field (see table x3) specifies the size and format of the extension parameters that follow in the XCDB descriptor.

**Table x3 — EXTENSION TYPE field**

Code	Descriptor Order <sup>a</sup>	Description	Extension size (bytes)	Reference
00h	first	CbCS Extension	tbd	tbd
01h	second	SA Security Extension	tbd	tbd
All others	Reserved			

<sup>a</sup> The order in which XCDB descriptors appear in an XCDB is arranged so that all the XCDB descriptors that follow an XCDB descriptor defined in a future version of this standard are also XCDB descriptors defined in a future version of this standard (i.e., after encountering one unrecognized XCDB descriptor, all subsequent XCDB descriptors are also going to be unrecognized).

{{The entries in the above table are only examples. The table is to be incorporated with all code values reserved.}}

The extension parameters contain additional information whose processing is associated with the CDB in the CDB field. The number, content, and size of the extension parameters depends on the contents of the EXTENSION TYPE field.

...

### 4.3.5 Common CDB fields

#### 4.3.5.1 Operation code

The first byte of a SCSI CDB shall contain an operation code identifying the operation being requested by the CDB. Some operation codes provide for modification of their operation based on a service action (see 4.3.4.2). In such cases, the operation code and service action code combine to identify the operation being requested. The location of the SERVICE ACTION field in the CDB varies depending on the operation code value.

The OPERATION CODE (see table 10) of the CDB has a GROUP CODE field and a COMMAND CODE field. The three-bit GROUP CODE field provides for eight groups of command codes. The five-bit COMMAND CODE field provides for thirty-two command codes in each group. A total of 256 possible operation codes exist. Operation codes are defined in this standard and other command standards (see 3.1.17). The group code value (see table 11) shall determine the length of the CDB.

**Table 10 — OPERATION CODE byte**

<b>Bit</b>	7	6	5	4	3	2	1	0
	GROUP CODE			COMMAND CODE				

The value in the GROUP CODE field specifies one of the groups shown in table 11.

**Table 11 — Group Code values**

Group Code	Meaning	Typical CDB format
000b	6 byte commands	see table 3 in 4.3.2
001b	10 byte commands	see table 4 in 4.3.2
010b	10 byte commands	see table 4 in 4.3.2
011b	reserved <sup>a</sup>	
100b	16 byte commands	see table 6 and table 7 in 4.3.2
101b	12 byte commands	see table 5 in 4.3.2
110b	vendor specific	
111b	vendor specific	

<sup>a</sup> The format of the commands using the group code 011b and operation code 7Fh is described in 4.3.3. [The format of the commands using the group code 011b and operation code 7Eh is described in 4.3.4.](#) With the exception of operation ~~code~~ codes 7Fh and 7Eh, all group code 011b operation codes are reserved.