

# ENDL TEXAS

Date: 18 March 2003  
 To: T10 Technical Committee  
 From: Ralph O. Weber  
 Subject: What is architectural about SAM-3 CDB definitions?

Almost since I started editing SAM I have wanted to move anything not directly part of the architecture from SAM to SPC. It probably is just as well that acting on that desire has taken so long because the tests of time have shown that almost everything in SAM-3 belongs there because it is part of the architecture or because it is intimately related to some part of the architecture.

For example, it might be argued that the status code value definitions are not part of the architecture, but in fact they are. The architecture is very concerned with which status to return when and so many aspects of task management are called into play when CHECK CONDITION status is returned that SAM would not be SAM without the status code definitions.

There is, however, one small corner of SAM-3 that does not have to be part of the architecture, the CDB format definition. As far as the architecture is concerned, the CDB communicates the command from application client to device server. T10 could completely reformat the CDB and the rest of the architecture could remain unaffected. As if to seal the deal on this, part of the CDB format definition is in SAM-3 clause 5 while the rest of it is in SPC-3 clause 4. Only two parts of the CDB are of any interest to the architecture:

- The OPERATION CODE field - If there is a command then there must be an operation code; and
- The CONTROL byte - Careful inspection shows that the CONTROL byte is the architecture's wholly owned field in the CDB, exercising control over linked commands and ACA (serious architectural task management stuff).

The objective of this proposal is moving CDB formatting issues that are not architectural from SAM-3 to SPC-3. Green is used to indicate text that is moving, blue indicates new text or text deleted for readability, purple indicates currently replicated text that is not needed in both standards.

## Revision History

- r0 Initial proposal
- r1 Changes approved by the March CAP working group

## r1 Discussion

During the March CAP working group, I became concerned about the following sentence in the revised SAM-3: "All CDBs shall have an OPERATION CODE as the first byte." I was mistakenly thinking that the sentence referred to the SERVICE ACTION field that has a different location in the variable length CDB format. I asked the group to recommend an r1 to fix this supposed error.

Clearly, the OPERATION CODE field is always in the first byte of a CDB, including in the variable length CDB format. The concern raised during the CAP working group was wrong.

Other than this descriptive text, r1 of this proposal is identical to r0.

## Specific Changes

All proposed changes reference SAM-3 r04 and SPC-3 r10.

**Change 1 [SAM-3 5.2 Command descriptor block (CDB)]:** Rewrite the subclause as follows:

**5.2 Command descriptor block (CDB)**

**5.2.1 CDB format**

The CDB defines the operation to be performed by the device server. ~~For some commands, the CDB is accompanied by a list of command parameters contained in the Data Out Buffer defined in 5.1. The parameters required for each command are specified in the applicable SCSI command standards.~~

~~If a logical unit validates reserved CDB fields and receives a reserved field within the CDB that is not zero or receives a reserved CDB code value, the logical unit shall terminate the command with CHECK CONDITION status; the sense key shall be set to ILLEGAL REQUEST with an additional sense code of INVALID FIELD IN CDB (see SPC-2).~~

For all commands, if the logical unit detects an invalid parameter in the CDB, then the logical unit shall complete the command without altering the media information.

All CDBs shall have an OPERATION CODE as the first byte. ~~All CDBs, except the CDB for operation code 7Fh, shall have a CONTROL byte as the last byte. The format for the CDBs with operation code 7Fh is defined in SPC-2.~~

Some operation codes provide for modification of their operation based on a service action. In such cases, the combination of operation code value and service action code value may be modeled as a single, unique command determinate. The location of the SERVICE ACTION field in the CDB varies depending on the operation code value.

~~The general format for all CDBs except the CDB for operation code 7Fh is shown in table 20. The remaining parameters depend on the command to be processed. All SCSI transport protocol standards shall accept CDBs less than or equal to 16 bytes in length. CDBs using the format shown in table 20 shall not exceed sixteen bytes in length.~~

**Table 20—Command Descriptor Block (CDB) Format**

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE							
1	Command specific parameters							
n-1								
n	CONTROL							

**5.2.2 OPERATION CODE byte**

~~The first byte of a SCSI CDB shall contain an operation code. The OPERATION CODE (see table 21) 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 the SCSI command standards. The group code value shall determine the length of the CDB (see table 22).~~

Table 21 — OPERATION CODE byte

Bit	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 22.

Table 22 — Group Code values

Group Code	Meaning
000b	6-byte commands
001b	10-byte commands
010b	10-byte commands
011b	reserved <sup>a</sup>
100b	16-byte commands
101b	12-byte commands
110b	vendor-specific
111b	vendor-specific
<sup>a</sup> The format the commands using the group code 011b and operation code 7Fh is described in SPC-2. With the exception of operation code 7Fh, all group code 011b operation codes are reserved.	

5.2.3 CONTROL byte

The CONTROL byte is the last byte of every CDB, except for the CDB for operation code 7Fh. The CONTROL byte is defined in table 23. All CDBs shall have a CONTROL byte (see table 23). The location of the CONTROL byte within a CDB depends on the CDB format (see SPC-3).

Table 23 — CONTROL byte

Bit	7	6	5	4	3	2	1	0
	Vendor specific		Reserved			NACA	Obsolete	LINK

Reset of Subclause Unchanged

Change 2 [SPC-3 4.3 The Command Descriptor Block (CDB)]: Rewrite the subclause as follows:

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 or receives a reserved CDB code value, the logical unit shall terminate the command with CHECK CONDITION status; the sense key shall be set to ILLEGAL REQUEST with an additional sense code of INVALID FIELD IN CDB (see SPC-2).

The fixed length CDB formats are described in 4.3.2. The variable length CDB formats are described in 4.3.3. The CDB fields that are common to most commands are described in 4.3.4. The fields shown in 4.3.2 and 4.3.3 and described in 4.3.4 are used consistently by most commands. However, the actual usage of any field (except OPERATION CODE and CONTROL) 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, it shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE.

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

4.3.2 The fixed length CDB formats **Unchanged**

4.3.3 The variable length CDB formats **Unchanged**

4.3.4 Common CDB fields

4.3.4.1 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 combination of operation code value and service action code value may be modeled as a single, unique command determinate. The location of the SERVICE ACTION field in the CDB varies depending on the operation code value.

The OPERATION CODE (see table x1) 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 the SCSI command this standard and other command standards (see 3.1.9). The group code value shall determine the length of the CDB (see table x2).

Table x1 — OPERATION CODE byte

Bit	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 x2.

**Table x2 — Group Code values**

Group Code	Meaning
000b	6 byte commands
001b	10 byte commands
010b	10 byte commands
011b	reserved <sup>a</sup>
100b	16 byte commands
101b	12 byte commands
110b	vendor specific
111b	vendor specific
<sup>a</sup> The format the commands using the group code 011b and operation code 7Fh is described in <a href="#">4.3.3 SPC-2</a> . With the exception of operation code 7Fh, all group code 011b operation codes are reserved.	

~~The OPERATION CODE field contains the code value identifying the operation being requested by the CDB. SAM-2 defines the general structure of the operation code value.~~ The OPERATION CODE field has a consistently defined meaning across all commands. This standard specifies the operation code values used by the commands defined herein.

#### 4.3.4.2 Service action **Unchanged**

#### 4.3.4.3 Logical block address **Unchanged**

#### 4.3.4.4 Transfer length **Unchanged**

#### 4.3.4.5 Parameter list length **Unchanged**

#### 4.3.4.6 Allocation length **Unchanged**

#### 4.3.4.7 Control

The contents of the CONTROL field are defined in [SAM-3](#) ~~SAM-2~~. The CONTROL field has a consistently defined meaning across all commands.