

Date: September 30, 2003
 To: T10 Committee (SCSI)
 From: Jim Coomes (Seagate)
 Subject: SBC/SPC ASC/ASCQ for Protection Information ~~Read~~ Error

1 Revised

This revision incorporates comments made at the Sept CAP meeting. The definition is changed to drop Read so the new error code may be used for Read and Write. The proposed sense key is changed to 0Bh (ABORTED COMMAND).

2 Overview

The following is a proposal from Richie Lary for a new error code for protection information read errors:
 The purpose of the special ASC/ASCQ for a bad DIF is twofold:

1) If you get a real HARD DISK ERROR condition, you can't trust the contents of the data buffer - its a "best guess" at the data. So you can't trust that the special DIF coding you used to flag special conditions (forced data error, uninitialized data, what have you) is accurate. The new ASCQ has the meaning of "this data passed all the usual disk checks but the DIF field didn't compare correctly", so you have more faith that any special coding you see in the buffer is really a special code and not an artifact. Without the new ASCQ you would have to reread the sector with DIF checking turned off to discriminate between true disk errors and special DIF codes, this hurts performance (and is not foolproof, either).

2) Many initiators keep track of hard error counts on the targets they access, and if they see an excessive error count they take some action, like notifying the service organization to replace the bad device. You don't want spurious HARD DISK ERROR ASCQ codes to trigger this mechanism on a perfectly healthy disk just because you used a special DIF encoding in your software!

The most common use for a special DIF coding that I'm aware of is as a "forced data error" flag. Imagine you have written some mirroring software, so you have two SCSI LUNs with identical content. One of these LUNs fails, so you acquire a "spare" LUN and perform a rebuild - "resilvering the mirror", as its called. During the rebuild you encounter a HARD DISK ERROR (a real one!) on some sectors of the source LUN. What do you write into those sectors of the destination LUN? If you write the "best guess" data with no error indication, you will eventually read this bad data and give it to a user without notifying them that it is bad - this is a major no-no. Many mirroring applications today use WRITE LONG to write a deliberate error onto the destination LUN, but this is device dependent and also has the property that it makes the destination disk look like it has a hardware problem when it doesn't. The ideal solution would be to write a special coding to the destination disk sector that says "the data in this sector is suspect" and put a good ECC on it - then you won't lie to the user about the data validity, but you won't cause the destination disk drive to appear defective in the process either. A special DIF coding (e.g. XOR hex "BADB" into the CRC) does the job very well.

Whether or not the data transfer continues after the bad DIF is detected is controlled by some existing check condition mode pages, I believe (I'm really not expert on the SCSI spec). Certainly that is the desired mode for the above application, to keep performance up...

2 SPC Change

A new ASC/ASCQ code, PROTECTION INFORMATION ~~READ~~ ERROR, is requested in SPC. Suggest 0Bh (ABORTED COMMAND) ~~03 (MEDIUM ERROR)~~ for a sense key and 47XXh (it would appear that 06h is next in the 47h family).

3 Document 03-176r54 Changes (SBC?)

The following changes are proposed to proposal 03-176r54 in order to add the new sense code.

Change

MEDIUM ERROR with the additional sense code set to UNRECOVERED READ ERROR

To:

ABORTED COMMAND with additional sense code set to PROTECTION INFORMATION ERROR

Change:

MEDIUM ERROR with the additional sense code set to PROTECTION INFORMATION WRITE ERROR

To:

ABORTED COMMAND with additional sense code set to PROTECTION INFORMATION ERROR