T10/99-208r2

To: T10 Technical Committee
From: Rob Elliott, Compaq Computer Corporation (Robert.Elliott@compaq.com)
Date: 3 November 1999
Subject: MRIE behavior with TEST bit asserted

There was a slight difference in the SPC-2 and SSC revision 17 definitions of the Method for Reporting Informational Exceptions (MRIE) bit in the Informational Exceptions mode page 02h. (MMC-2 includes portions of the SPC-2 description)   See SPC-2 revision 10 section 8.3.6 and SSC revision 17 section 5.3.4.6. SSC will drop all the text that restates SPC-2, so it will no longer have different behavior. However, some changes are requested in SPC-2 based on the SSC text, since it is more useful for test software.

For MRIE = 4, for example, SPC-2 says:
>   This method instructs the device server to report informational exception conditions, regardless of the value of the per bit of the error recovery parameters mode page, by returning a CHECK CONDITION status *on any command*.

SSC said:
>   This method instructs the device server to report information exception conditions (regardless of the value of the per bit of the error recovery parameters mode page) by returning a CHECK CONDITION status *on the next SCSI command (excluding Inquiry and Request Sense)* after an informational exception condition was detected.

There are two differences:
1. SSC exempts INQUIRY and REQUEST SENSE. SPC-2 does not.   This exemption text appears in SSC for MRIE values 2, 3, 4, and 5.

The only other way to cause INQUIRY and REQUEST SENSE to return CHECK CONDITION is to use a reserved bit. SSC's exemption seems appropriate. (The use of parenthesis, however, may be improper).

2. SSC says the CHECK CONDITION will be returned on the "next" command. SPC-2 says "any" command. This appears only in SPC's MRIE = 4 description, although it appears in all the SSC descriptions.

In case of an actual hardware exception condition, the application client cannot tell exactly when it happened, so this doesn't matter. Devices may only want to return an error only on commands which relate to the error. If the TEST bit is being used to trigger a fake exception condition, however, the SSC requirement is more useful since the command that must cause the CHECK CONDITION can be predicted.

There is little benefit in creating CHECK CONDITIONs on commands that will never generate them in real operation. However, it is important that test software can force a CHECK CONDITION to be generated for every command that could generate one in real operation.

Suggested changes
Editorial: Change "the error recovery parameters page" to "the error recovery page" to match the names in SSC and SBC. SBC has both a "read-write error recovery page" and a "verify error recovery page." SSC just has a "read-write error recovery page."

Technical: Make the "next" behavior mandatory for TEST mode exceptions, but let real events report exceptions after "any" command.

New SPC-2 Table 157 wording:

Table 157 — Method of reporting informational exceptions (~~MIRE~~MRIE) field (part 1 of 2)

3h - **Conditionally generate recovered error:** This method instructs the device server to report informational exception conditions, if the reporting of recovered errors is allowed, by returning a CHECK CONDITION status. If TEST = 0, the status may be returned on any command after the informational exception condition occurs. If TEST = 1, the status shall be returned on the next command that is normally capable of returning an informational exception condition when TEST = 0. The sense key shall be set to RECOVERED ERROR and the addi-tional sense code shall indicate the cause of the informational exception condition.

A command that has the CHECK CONDITION shall complete without error before any informa-tional exception condition may be reported.

4h - **Unconditionally generate recovered error:** This method instructs the device server to report informational exception conditions, regardless of the value of the per bit of the error recovery parameters mode page, by returning a CHECK CONDITION status ~~on any command~~. If TEST = 0, the status may be returned on any command after the informational exception condition occurs. If TEST = 1, the status shall be returned on the next command that is normally capable of returning an informational exception condition when TEST = 0. The sense key shall be set to RECOVERED ERROR and the additional sense code shall indicate the cause of the informational exception condition.

The command that has the CHECK CONDITION shall complete without error before any infor-mational exception condition may be reported.

5h - **Generate no sense:** This method instructs the device server to report informational exception conditions by returning a CHECK CONDITION status. If TEST = 0, the status may be returned on any command after the informational exception condition occurs. If TEST = 1, the status shall be returned on the next command that is normally capable of returning an informational exception condition when TEST = 0. The sense key shall be set to NO SENSE and the additional sense code shall indicate the cause of the informational exception condition.

The command that has the CHECK CONDITION shall complete without error before any infor-mational exception condition may be reported.