

**To:** T10 Technical Committee  
**From:** Rob Elliott, Compaq Computer Corporation  
(Robert.Elliott@compaq.com)  
**Date:** 16 November 1998  
**Subject:** Conflicting "reserved" definitions

There seems to be a problem in the SCSI-3 specifications with the definitions of "reserved." Most include a sentence like this:

The recipient may check reserved bits, bytes, words, or fields.

Some use "may check." Some use "may not check." Some use "shall not check." According to the T10 minutes from March, the SBC wording "may not check" was supposed to be used in all new standards.

As I understand it, "may" and "may not" have the same meaning in standards-speak. The meaning is always "might" rather than "shall." The "shall not" usage, however, changes the meaning. I suggest that "may or may not check" be used in the "reserved" definition of each new SCSI-3 specification.

Also, I suggest that a definition for "may not" be added to each glossary keywords section.

Example (from SPI-3):

**3.3.4 may:** A keyword that indicates flexibility of choice with no implied preference (equivalent to "may or may not").

**3.3.x may not:** A phrase that indicates flexibility of choice with no implied preference (equivalent to "may or may not").

### History

In SPI-2 ballot comments resolved in March, Gene Milligan made this comment:

(98-108r3)

32) The desire for recipients not to check for more recent, or inadvertent use, of reserved bits has been in place for a long while. Is it not time to be more blatant than "Recipients may check reserved bits, bytes, words or fields for zero values and report errors if non-zero values are received."?

Accepted: Wording changed to "Recipients may not check reserved bits, bytes, words or fields for zero values."

In the plenary, the inconsistency of "reserved" definitions across the SCSI-3 standards was noted, and T10 agreed to merge them:

(98-126r1)

In response to Milligan comment 32, it was agreed to copy the SBC wording to SPI-2, SPC-2, SAM-2, and other new standards containing a definition for the reserved keyword.

However, this was not done exactly:

- SBC had "may not check"
- SPC-2 and SAM-2 were changed to "may check"
- RBC and MMC-2 have "shall not check."

Here are some sample definitions of "reserved" from the various standards:

**SPC revision 11a original wording ("may check...and report errors")**

3.3.7 reserved: A keyword referring to bits, bytes, words, fields and code values that are set aside for future standardization. A reserved bit, byte, word or field shall be set to zero, or in accordance with a future extension to this standard. Recipients may check reserved bits, bytes, words or fields for zero values and report errors if non-zero values are received. Receipt of reserved code values in defined fields shall be reported as error.

**SBC revision 8 wording ("may not check"; supposedly the new standard)**

3.3.6 reserved: Refers to bits, bytes, words, fields, and code values that are set aside for future standardization. Their use and interpretation may be specified by future extensions to this or other standards. A reserved bit, byte, word, or field shall be set to zero, or in accordance with a future extension to this standard. The recipient may not check reserved bits, bytes, words, or fields. Receipt of reserved code values in defined fields shall be treated as an error.

**SPI-2 revision 20a new wording ("may not check")**

3.3.6 reserved: A keyword referring to bits, bytes, words, fields and code values that are set aside for future standardization. A reserved bit, byte, word or field shall be set to zero, or in accordance with a future extension to this standard. Recipients may not check reserved bits, bytes, words or fields for zero values. Receipt of reserved code values in defined fields shall be reported as error.

**SPC-2 revision 5**

**SAM-2 revision 9 ("may check")**

3.3.8 reserved: A keyword referring to bits, bytes, words, fields and code values that are set aside for future standardization. A reserved bit, byte, word or field shall be set to zero, or in accordance with a future extension to this standard. Recipients may check reserved bits, bytes, words or fields for zero values. Receipt of reserved code values in defined fields shall be reported as error.

**CAM-3 revision 3 ("should not check")**

3.1.18 reserved

Where this term is used for bits, bytes, fields, and code values; the bits, bytes, fields, and code values are set aside for future standardization. The default value shall be zero. The

originator is required to define a reserved field or bit as zero, but the receiver should not check reserved fields or bits for zero.

**MMC-2 revision 8 ("shall not check")**

3.4.8. reserved - A keyword referring to bits, bytes, words, fields and code values that are set aside for future standardization. Their use and interpretation may be specified by future extensions to this or other standards. A reserved bit, byte, word, or field shall be set to zero, or in accordance with future extension to this standard. The recipient shall not check reserved bits, bytes, words or fields. Receipt of reserved code values in defined fields shall be treated as an error.

**RBC revision 5 (“shall not check”)**

3.1.6 reserved: A keyword used to describe objects—bits, bytes, and fields—or the code values assigned to these objects in cases where either the object or the code value is set aside for future standardization. Usage and interpretation may be specified by future extensions to this or other standards. A reserved object shall be zeroed or, upon development of a future standard, set to a value specified by such a standard. The recipient of a reserved object shall not check its value. The recipient of a defined object shall check its value and reject reserved code values.

A sample older standard:

**SBP-2 revision 4 ("shall not check")**

3.1.1.4 reserved: A keyword used to describe bits, bytes, quadlets, octlets and or the code values assigned to these objects in cases where either the object or the code value is set aside for future standardization. Usage and interpretation may be specified by future extensions to this or other standards. A reserved object shall be zeroed or, upon development of a future standard, set to a value specified by such a standard. The recipient of a reserved object shall not check its value. The recipient of an object defined by this standard as other than reserved shall check its value and reject reserved code values.