

ENDL TEXAS

Date: 22 September 2001
 To: T10 Technical Committee
 From: Ralph O. Weber
 Subject: Replacing Task Identifiers in SAM-2

The 30 April SAM-2 editing meeting reached the conclusion that the term 'task identifier' could be replaced with I_T_L_x Nexus, I_T_L_Q Nexus, or I_T_L Nexus, depending on the context. It was further noted that the usage of 'task identifier' was for the most part confined to subclause 4.9, further suggesting that using a more familiar term (e.g., Nexus) would clarify the description. However, these changes were not made for two reasons. They were thought to go slightly beyond the bounds of editorial changes and it was felt that careful research was needed to ensure that the change was implemented properly. This proposal details the changes needed to change the usage of 'task identifier' to some form of I_T_L_x Nexus.

When 01-196r0 was presented to the July CAP working group meeting, requests were made to reconsider the wording in light of the fact that a nexus is a relationship not an address. 01-196r1 (this revision) responds to those concerns.

Strikethrough indicates text to be removed and underline indicates text to be added. All PDF page number references are to sam2r20.pdf.

Change 1: Subclause 3.1.58 PDF page 31

Modify the definition of 'linked command' as indicated:

linked command: One in a series of SCSI commands processed by a single task that collectively make up a discrete I/O operation. In such a series, each command ~~has is represented by~~ the same ~~task identifier~~ I_T_L_x nexus, and all, except the last, have the LINK bit in the CDB CONTROL byte set to one.

Change 2: Subclause 3.4 PDF page 37

Modify the description of callable procedure conventions as indicated:

For instance, ~~Task Identifier~~ Sense Data is the name of an argument in the **Execute Command** procedure call.

Note that this sentence needs to be modified regardless of the disposition of this proposal because Task Identifier no longer is an argument in the **Execute Command** procedure call.

Change 3: Subclause 4.9 PDF pages 53

The changes required in subclause 4.9 are so extensive that the entire subclause is presented here as a single unit with modifications marked throughout. Note that editor's note 3 should be deleted although that is not shown.

4.9 Tasks

4.9.1 The task object

The task object represents either a tagged task or an untagged task. The composition of a task includes a definition of the work to be performed by the logical unit in the form of a command or a group of linked commands. A tagged task ~~is represented by~~ an I_T_L_Q nexus (see 4.10) ~~and~~ is composed of a definition of the work to be performed by the logical unit, ~~a tagged task identifier (see 4.9.3)~~, and a task attribute (see 7.6). An untagged task ~~is represented by~~ an I_T_L nexus (see 4.10) ~~and~~ is composed of a definition of the work to be performed by the logical unit, ~~a untagged task identifier (see 4.9.3)~~, and implicitly a SIMPLE task attribute (see 7.6). ~~Task identifier (see 4.9.3) refers to either a tagged task identifier or an untagged task identifier.~~

~~The I_T_L_Q nexus representing a A~~ tagged task includes a tag (see 4.9.2) ~~in its tagged task identifier that allows allowing~~ many uniquely identified tagged tasks to be present concurrently in a single task set. A tagged task also includes one of the task attributes described in 7.6 that allows the initiator to specify processing relationships between various tagged tasks. An untagged task does not include a tag in ~~its I_T_L nexus any of its component definitions~~, thus restricting the number of concurrent untagged tasks in a single task set to one per initiator. Also, an untagged task is assumed to have a SIMPLE task attribute, leaving the initiator no control over its relationship to other tasks in the task set.

Every SCSI protocol shall support tagged and untagged tasks. Support for tagged tasks by a logical unit is optional.

~~A task identifier An I_T_L_x nexus~~ that is in use shall be unique as seen by the initiator originating the command and the logical unit to which the command was addressed. ~~A task identifier An I_T_L_x nexus~~ is in use over the interval bounded by the events specified in 5.5). ~~A task identifier An I_T_L_x nexus~~ is unique if one or more of its components is unique within the ~~scope specified above specified time interval~~. By implication, therefore, an initiator shall not cause the creation of more than one untagged task having identical values for the target identifier and logical unit number. ~~Conversely An~~ initiator may create more than one task with the same tag value, provided at least one of the remaining ~~task identifier I_T_L_x nexus~~ components is unique.

4.9.2 Task tags

A tag is a field containing up to 64 bits that is a component of ~~a tagged task identifier an I_T_L_x nexus~~. An initiator assigns tag values in each ~~tagged task identifier I_T_L_Q nexus~~ in a way that ensures that the ~~identifier nexus~~ uniqueness requirements stated in ~~4-9 4.9.1~~ are met.

4.9.3 Identification of tasks

~~A device server identifies a task with a task identifier. The task identifier object represents either a tagged task identifier or an untagged task identifier. A tagged task identifier is composed of an initiator identifier (see 4.7.1), a logical unit number (see 4.8) and a tag (see 4.9.2). An untagged task identifier is composed of an initiator identifier and a logical unit number.~~

The above changes were presented in 01-169r0 and have been revised as requested by the July CAP working group. The following additional changes are proposed to make other uses of I_T_L_x nexus suggest a relationship.

Change 4: Subclause 5.8.2 PDF pages 88

An overlapped command occurs when a task manager detects ~~the use of~~ a duplicate I_T_L_x nexus (see 4.9.1) in a command before a pending task holding that I_T_L_x nexus completes its task lifetime (see 5.5).