

Document: T10/00-354r1  
To: T10 Committee Membership  
From: Edward A. Gardner, Ophidian Designs  
Date: 1 November 2000  
Subject: 64-bit Tags in SVP

When discussing SVP with the IBTA Application Working Group, several members requested that SVP commands and responses include a 64-bit unique identifier. The AWG requested that I propose this to T10.

The desire is that every SVP command IU contain a 64-bit identifier that would be returned in the corresponding response IU. Host driver software would use this to uniquely identify all outstanding IO operations across all Initiators (adapters), Targets and LUNs. SVP's current 32-bit Task Tags were intended to allow this, but members of AWG felt that field size would be too small for future large scale systems.

As described in revision 0 of this document, the simplest way to accomplish this is to increase SVP's TAG field size to 64 bits. That change was approved at the T10 CAP Working Group on September 13, 2000 and at the T10 Plenary meeting on September 14, 2000.

However, that working group meeting also indicated a strong desire to take advantage of the 64-bit TAG being unique among all requests from an Initiator, not just within an I\_T\_L nexus as (minimally) mandated by SAM. This allows the LUN field to be removed from the SVP\_RSP information unit, and repositioned within the SVP\_CMD information unit to better match FCP. Revision 1 of this document describes the changes to incorporate that recommendation into SVP.

I encountered two questions while drafting revision 1 that were not clear from the previous working group discussion. The first is whether targets check or enforce that the TAG field value is unique. One approach is to specify that initiators are **expected** to supply unique TAG field values and that targets **shall not** check whether the values are unique. That is the wording I propose below. An alternative approach would be to specify that initiators **shall** supply unique TAG field values and that targets **shall** check whether the values are unique.

The other issue is the ABORT TASK request. Specifically, within an SVP\_CMD information unit that contains an ABORT TASK request, which field contains the TAG of the task to be aborted? The seemingly obvious choice that the normal TAG field contains the TAG value to be aborted results in ABORT TASK being an exception. The resulting description of the TAG field would read something like:

**The TAG value of an SVP\_CMD request containing an ABORT TASK task management function shall match the TAG value of the SVP\_CMD request that contained the task to be aborted. An initiator is expected to provide a TAG value in other requests that is unique among all of the initiator's outstanding requests.**

This exception would be reflected in the SVP driver code. When an ABORT TASK is issued, sometimes the initiator will first receive an ABORT TASK response, sometimes a response to the task that was to be aborted (if the target sent the task response before it received the ABORT TASK). The initiator must distinguish these two responses, as in the latter case the ABORT TASK response is still pending and the initiator cannot re-use the TAG value (and associated data structures) until after that response arrives.

Unfortunately, as presently defined for SPI-n and FCP-n, the responses for a successfully completed task and a successfully completed task management request are identical. While it is necessary for the initiator to distinguish these two responses, at present there is no way to do so. Note that FCP-n does not have this problem because it uses an FC-2 layer abort exchange operation for ABORT TASK rather than an information unit. SPI-n does not have this problem because it is an interlocked bus.

We could solve this problem by inventing a unique status or RESPONSE DATA code to indicate successful completion of an ABORT TASK request. The approach I prefer and have described below is to assume that the TAG field is unique for all requests, without an exception for ABORT TASK. The TAG of the task to be aborted would be conveyed in some other field of an ABORT TASK request. The first eight bytes of the CDB field seem the obvious choice.

The remainder of this document describes the proposed changes to SVPr01.

Table 1 replaces Table 2 in SVPr01, it summarizes the changes for all IUs. The order of the REQUESTLIMITDELTA and TAG fields is reversed, and TAG extended to eight bytes. Note that bytes 12-15 were formerly reserved in every IU, they are now part of the TAG field.

**Table 1 - Fields common to all information units**

Bit Byte	7	6	5	4	3	2	1	0	
0	TYPE								
1									
2	RESERVED								
3									
4	MSB	REQUESTLIMITDELTA							
...	(ONLY WHEN SENT BY TARGET)								
7	LSB								
8									
...	TAG								
15									
16									
...	varies								
n									

Replace the last two paragraphs of clause 5.1 with the following:

Bytes 4 through 7 of each IU sent by an SVP target contain REQUESTLIMITDELTA. See clause 4.3 for a description of that field's use. Those bytes are used for other purposes in IUs sent by an SVP initiator.

Bytes 8 through 15 of each IU contain a TAG value, which provides a mechanism for matching requests with their corresponding responses. A requestor is expected to provide a TAG value in each request that is unique among all of the requestor's outstanding requests. A responder shall copy the TAG value from each request to its response. Responders shall not check whether the TAG values of outstanding requests are unique.

Table 2 replaces Table 3 in SVPr01, it shows the revised SVP\_CMD IU format. In addition to the TAG field changes described above, the LOGICAL UNIT NUMBER field has been moved to the same position (relative to the CDB) as in FCP.

Delete the fifth paragraph from clause 5.2 (description of TAG field in SVP\_CMD IU).

Replace the twelfth paragraph of clause 5.2 (located between Table 4 and Table 5) with the following:

The TASK MANAGEMENT FLAGS field is defined in table 5. If TASK MANAGEMENT FLAGS specifies a task management function, DATA LENGTH, TASK ATTRIBUTE, ADDITIONAL CDB LENGTH, RDDATA and WRDATA shall contain zero; ADDITIONAL CDB shall not be present. If TASK MANAGEMENT FLAGS specifies an ABORT TASK task management function, the first 8 bytes of the CDB field shall contain the TAG value of the task to be aborted; the remainder of the CDB field shall contain zero. If TASK MANAGEMENT FLAGS specifies any other task management function, the entire CDB field shall contain zero.

Table 3 replaces Table 6 in SVPr01, it shows the revised SVP\_RSP IU format. In addition to the TAG and REQUESTLIMITDELTA field changes described above, the LOGICAL UNIT NUMBER field has been deleted.

Table 2 - SVP\_CMD information unit

Bit Byte	7	6	5	4	3	2	1	0
0	TYPE							
1								
2	RESERVED							
7								
8								
...	TAG							
15								
16	MSB							
...	DATA VIRTUAL ADDRESS							
23								LSB
24	MSB							
...	DATA MEMORY HANDLE							
27								LSB
28	MSB							
...	DATA LENGTH							
31								LSB
32								
...	RESERVED							
35								
36	MSB							
...	LOGICAL UNIT NUMBER							
43								LSB
44	RESERVED							
45	RESERVED				TASK ATTRIBUTE			
46	TASK MANAGEMENT FLAGS							
47	RESERVED	ADDITIONAL CDB LENGTH = (n-63)/4				RdDATA	WRDATA	
48	MSB							
...	CDB							
63								LSB
64	MSB							
...	ADDITIONAL CDB							
n								LSB

Table 3 - SVP\_RSP information unit

Bit	7	6	5	4	3	2	1	0	
Byte									
0	TYPE								
1									
2	RESERVED								
3									
4	MSB								
...	REQUESTLIMITDELTA								
7								LSB	
8									
...	TAG								
15									
16	RESERVED								
17									
18	RESERVED			RIDUNDER	RIDOVER	SNSVALID	RSPVALID		
19	STATUS								
20	MSB								
...	RESIDUAL COUNT								
23								LSB	
24	MSB								
...	SENSE DATA LIST LENGTH = n								
27								LSB	
28	MSB								
...	RESPONSE DATA LIST LENGTH = m								
31								LSB	
32	MSB								
...	RESPONSE DATA (m bytes long)								
31+m								LSB	
32+m	MSB								
...	SENSE DATA (n bytes long)								
31+m+n								LSB	