

DocNum: T10/01-057r0  
Date: 2001/01/23  
Author: Eric Anderson (ewa@apple.com)  
Title: **Proposal for Fast Start in SBP-3**

## Summary

This document proposes an addition to SBP-3 called Fast Start. This new feature provides an optional way to more quickly start the execution of an ORB by a fetch agent that is in the SUSPENDED or RESET state, in a manner that is fully backwards-compatible with existing ORB services.

Practical experience with SBP-2 disk drives has shown that it is frequently the case that the fetch agent is in the SUSPENDED state, the initiator is aware of this, and the initiator desires to initiate new activity. Under the most favorable conditions for this case, three packets must be exchanged on Serial Bus before payload can be moved for a new ORB. These are a write to ORB\_POINTER, a read of the ORB, and a read response. Under the least favorable conditions, seven packets must be exchanged. These are a write to DOORBELL, a read of the old ORB, a read response, a read of the new ORB, another read response, a read of the page table, and a third read response. This proposal describes a Fast Start packet that can be used to start new activity. When this packet is employed, only one packet crosses Serial Bus prior to payload transfer.

The Fast Start packet combines three data structures into one: An ORB\_POINTER, an ORB, and part or all of a page table. An initiator may send a Fast Start packet to a target in situations when it would be safe to write the ORB\_POINTER register directly. See section 9.1.3, and consider that in modern use the behavior described therein is not limited to the BIOS, because it yields a performance improvement in contemporary systems (relative to the behavior described in 9.1.2) whenever the initiator is aware that the target is in the SUSPENDED state. When the target receives a Fast Start packet, the target has all the information needed to begin payload transfer immediately. In the case of a large transfer, the target may need to fetch additional page table information, but this can be done in parallel with payload transfer.

Because the use of the Fast Start packet is governed by the same conditions as described in 9.1.3, and because the execution of the ORB and the resulting target state are unchanged from SBP-2, Fast Start is fully backwards-compatible and requires minimal modification of the target or initiator. Additionally, the Fast Start packet can only be used when both the initiator and the target are willing to do so; in any other situation the behavior reverts to the unmodified standard model. Consequently, targets or initiators that do not implement Fast Start incur

no burden from this proposed enhancement, and targets and initiators that do implement Fast Start remain fully interoperable with those that do not.

In this proposal, the ability to support Fast Start is indicated in the target by way of a new Configuration ROM key called SBP\_Features. The initiator does not need to indicate its support for Fast Start; initiators that do support Fast Start will detect similarly capable targets and then employ Fast Start at their discretion. The use of the SBP\_Features key enables the implementation of Fast Start without modifying any existing SBP data structures.

The remainder of this document describes changes and additions that could be made to the SBP-3 draft standard (Revision 1 of January 5, 2001) to incorporate Fast Start.

#### **5.4 Fast Start packet**

(New section)

The Fast Start packet is formed by concatenating three SBP data structures defined elsewhere in this document: An ORB\_POINTER (see 6.4.3), an ORB (see 5.1), and (optionally) a page table (see 5.2). The Fast Start packet is assembled by the initiator and sent to the target using a Serial Bus block write transaction.

The target needs an ORB\_POINTER so that it can report status as defined in section 5.3, even though the entire ORB is included in the Fast Start packet. The ORB\_POINTER may also be read by the initiator as described in section 6.4.3. The ORB included in the Fast Start packet shall exist in initiator memory at the address indicated by ORB\_POINTER.

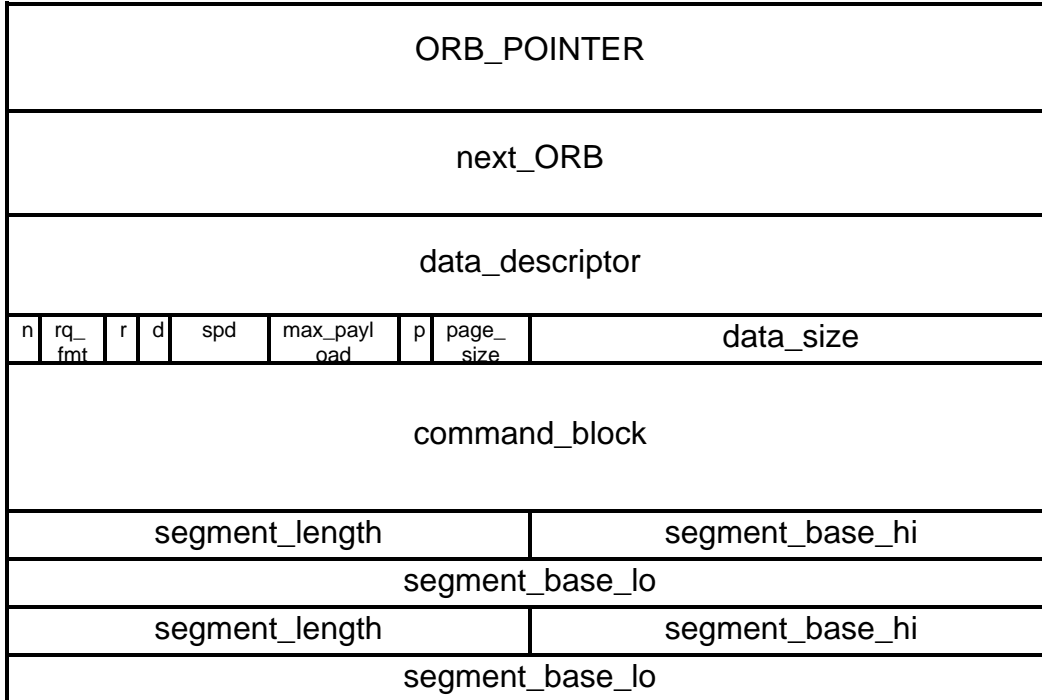
If an ORB does not reference a page table, no page table is included in a Fast Start packet for that ORB. The page table included in a Fast Start packet (if any) shall exist at the address indicated by data\_descriptor.

If a protocol transported by SBP employs variable-sized ORBs, the target is responsible for examining the ORB within the Fast Start packet to determine where the boundary between the ORB and the page table (if any) is located. How this determination is made is beyond the scope of this standard.

If a partial page table is included in a Fast Start packet, the target is responsible for examining the total length of the packet to determine how much of the page table has been provided. The target must then fetch the remainder of the page table (possibly in multiple reads) prior to completing the ORB.

A Fast Start packet shall always be sent by the initiator in a single Serial Bus block write packet.

The following figure shows an example of a Fast Start packet for an ORB that includes 12 bytes of command\_block and references a small page table.



#### 6.4 Command block and stream control agent registers

(Add paragraph at end of existing section 6.4)

A target may optionally support a FAST\_START register at an address indicated in the SBP\_Features key. The FAST\_START register supports the receipt of a block write packet as shown in section 5.4.

##### 6.4.6 FAST\_START register

(New section)

The FAST\_START register can receive a block write packet formatted as shown in section 5.4. The FAST\_START register is located at an address determined by the SBP\_Features key (see 7.4.13).

#### 7.4.13 SBP\_Features key

(New section)

The SBP\_Features entry is an immediate entry that, when present in the unit directory, specifies the optional SBP features that apply to all logical units within the target. The following figure shows the format of this entry.

|                  |          |     |        |
|------------------|----------|-----|--------|
| 3E <sub>16</sub> | reserved | fsf | offset |
|------------------|----------|-----|--------|

The fast\_start\_feature field (abbreviated as fsf in the figure above) specifies whether the target or logical unit supports the Fast Start feature, and if so, how many page table entries may be accepted in a Fast Start packet, according to the following encoding.

| fsf | Fast Start support                                |
|-----|---|
| 0   | No support for Fast Start                         |
| 1   | Up to 1 page table entry in Fast Start packet     |
| 2   | Up to 2 page table entries in Fast Start packet   |
| 3   | Up to 4 page table entries in Fast Start packet   |
| 4   | Up to 8 page table entries in Fast Start packet   |
| 5   | Up to 16 page table entries in Fast Start packet  |
| 6   | Unlimited page table entries in Fast Start packet |
| 7   | Reserved  |

The offset field specifies the offset in quadlets of the Fast Start register relative to the fetch agent base (as indicated in the login response packet).

### 7.5.5 SBP\_Features entry

(New section)

The SBP\_Features entry is an immediate entry that, when present in a logical unit directory, specifies the optional SBP features that apply to that logical unit.

### 9.1.4 Fetch agent use with Fast Start (informative)

(insert section before existing 9.1.4; renumber existing section to 9.1.5)

When the initiator is aware that the target is in the RESET or SUSPENDED state, the initiator may request the execution of a new ORB by writing a Fast Start packet (see 5.4) to the FAST\_START register (see 6.4.6). The initiator may know the target is in one of these two states for several possible reasons. The target may have most recently signaled the completion of a Login or Reconnect operation, proving that the target is in the RESET state. Or the target may have signaled the completion of a normal command ORB with a status block

containing a src value of 1 (see 5.3), proving that the target is in the SUSPENDED state.

Upon activation of the target by a Fast Start packet, the fetch agent shall continue as if the ORB had been signaled by a write to the ORB\_POINTER register, except that it should avoid redundant Serial Bus reads of ORB or page table data that was included in the Fast Start packet. In particular, the ORB shall store status normally upon completion of the ORB, if appropriate, and shall branch to any next\_ORB indicated in the ORB. If the ORB has no next\_ORB pointer, the target shall enter the SUSPENDED state upon completion of the ORB.

#### **Annex D (informative) Sample configuration ROM**

(expand to show example of SBP\_Features key)