To: T10 Technical Committee
From: Rob Elliott, HP (elliott@hp.com)
Date: 23 October 2003
Subject: 03-364r0 MSC Report Bridge Mapping command

Revision history
Revision 0 (23 October 2003) First revision

Related documents
03-354 SPC-3 Specify initiator ports in EXTENDED COPY target descriptors (Rob Elliott, HP)
03-353 SPC-3 Report initiator port identifiers (Rob Elliott, HP)
03-344 SPC-3 Report all target port identifiers (Rob Elliott, HP)
03-242 MSC presentation (George Penokie, IBM)
02-037 MSC Management commands proposal (Bob Griswold, Crossroads)
01-151 BCC project proposal issues (Rob Elliott, HP)
01-095 Project proposal for Management Server Commands (MSC) (Rob Elliott, HP)

Overview
Transport protocol bridges (aka. gateways or routers)(e.g. iSCSI to Fibre Channel) should to be visible to SCSI applications. This reduces or eliminates the need for bridges to intercept and change data in commands ranging from INQUIRY to EXTENDED COPY.

This information might be available via out-of-band management structures such as CIM, but it's hard to guarantee that the SCSI application has access to that data. If the the application can generate SCSI commands that go through the bridge, it is very likely that it can generate SCSI commands to the bridge itself.

Figure 1 shows an example iSCSI to FCP bridge. The SCSI device objects are not shown, because they are only evident in the INQUIRY command VPD data.

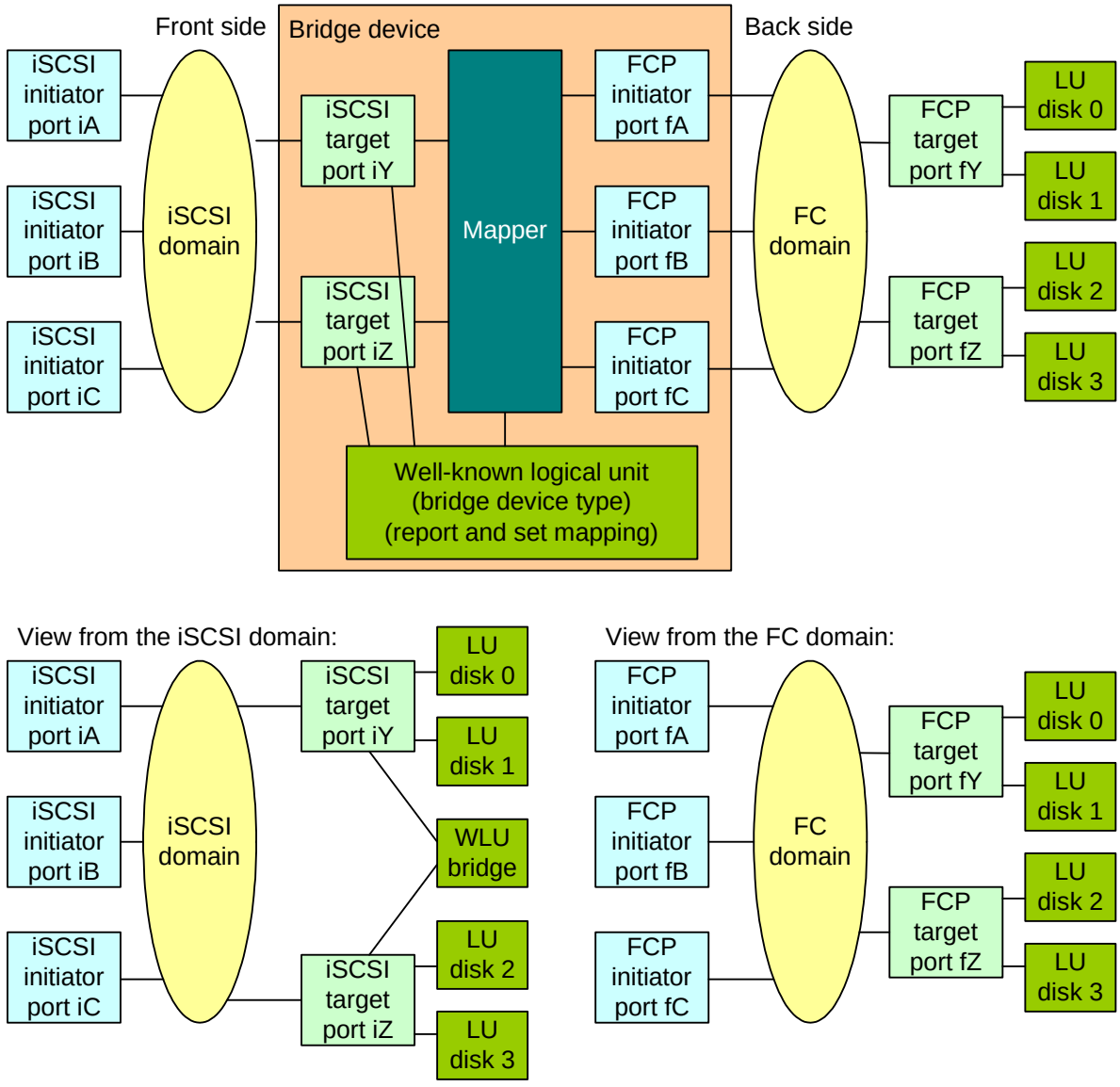

Figure 1 — iSCSI to FCP bridge

Bridges vary in complexity and capability. For back-to-front mapping of targets, they might map:

a) **Target ports**. Each back side target port (e.g. FC target port) is mapped to a unique front side target port (e.g. iSCSI target port). Logical unit numbers accessible via each target port are unchanged, even though the protocol of the target port apparently changes.

) This is the cleanest model, but may be difficult to implement on some protocols. In iSCSI and SRP, it's easy to create another target port, since they are already virtual constructs. In FCP, a virtual N_port needs to be used or the bridge needs to look like a switch with lots of N_Ports behind it. In SAS, the bridge needs to look like an expander so it can claim more than one SAS address; a SAS physical link to an end device is not allowed to support more than one SAS address.

b) **Logical units**. The bridge presents a single target port on the front side. Select LUNs from select back side target ports are mapped to LUNs behind the front side target port. Due to the likely collisions (most of the back-side LUNs are probably LUN 0), the logical unit numbers must be remapped (all but one of the LUN 0s has to map to a non-zero LUN). Since this bridge changes the target port/logical unit relationship, it introduces several problems:

A) REPORT LUNS data is wrong. The bridge must intercept REPORT LUNS (needs to return the front-side target port's LUN map, not the LUN map of some back side target port)
B) TARGET RESET is wrong. The bridge needs to reset all the peer LUs as seen on the front-side, not the peers of some back side LU. Care must be taken to reset all logical units that are mapped through the bridge, but not to reset those that are not mapped.
C) Commands using the relative target port identifier are wrong. Target Port Group commands in particular behave oddly.
D) Well-known logical units conflict with each other. Since they have fixed LUNs, they overlap with each other and with the bridge's own well-known logical units.

For front-to-back mapping of initiators, a bridge can do is:

a) **Initiator ports**. Each front side initiator port is mapped to a unique back side initiator port. This is critical for commands that care about initiator identity like Access Controls and Persistent Reservations.

) If the bridge is from a SAN protocol to parallel SCSI, there is a severe reduction in the number of initiators that can be supported on the parallel SCSI bus (1, 2, maybe 4). The bridge has to choose which initiators to allow access.

b) **None**. Some bridges just ignore this and present one initiator port on the back side. Commands cognizant of initiator identity like Persistent Reservations and Access Controls do not operate correctly unless the bridge intercepts them and implements them itself.

) These bridges should at least be identified so applications can tell not to send certain commands through them.

Bridge device type

Current protocol bridges use the SCSI Controller Commands (SCC) device type for their control logical units. So do RAID controllers, the type of devices intended for that device type. Neither bridges nor RAID controllers, however, comply with the SCC command set.

Rather than continue to misuse the SCC device type, a new SCSI device type is proposed for bridge devices. A SCSI target device with bridge functionality shall have at least one logical unit of the bridge device type. It may also have logical units containing copy managers, mapped logical units, etc. The bridge LU shall be accessible through every target port in the bridge device (possibly with a different LUN).

The bridge logical unit shall supports these commands:

a) **INQUIRY**: Returns PERIPHERAL DEVICE TYPE field set to Bridge.
b) **REPORT LUNS**: Reports the LUNs accessible via the front-side target port that is being used to access this logical unit.
c) **TEST UNIT READY**: Probably always GOOD status since there is no media to access. If bridge mapping tables ever get large and are stored on a disk, this could be used to indicate the mapping table is ready.
d) **REPORT BRIDGE MAPPING**: Returns a data structure describing all the mapping that a bridge is doing.
e) **WAIT FOR BRIDGE MAPPING CHANGE**: Since the SCSI architecture no longer defines asynchronous event notification, this serves as a camp-on command that completes and returns status whenever is change in mapping occurs.
f) **SET BRIDGE MAPPING TABLE**: Define in the future (if ever). MSC should start small and just define tools as they are needed.

The logical unit shall generate a unit attention whenever its mapping changes.

Addressing of nested bridges needs to be resolved. A well known logical unit works well for one bridge. However, if the first W-LU claims ownership of a W-LUN, The W-LUs of a bridge behind the bridge are unreachable. shows an example of nested bridges.

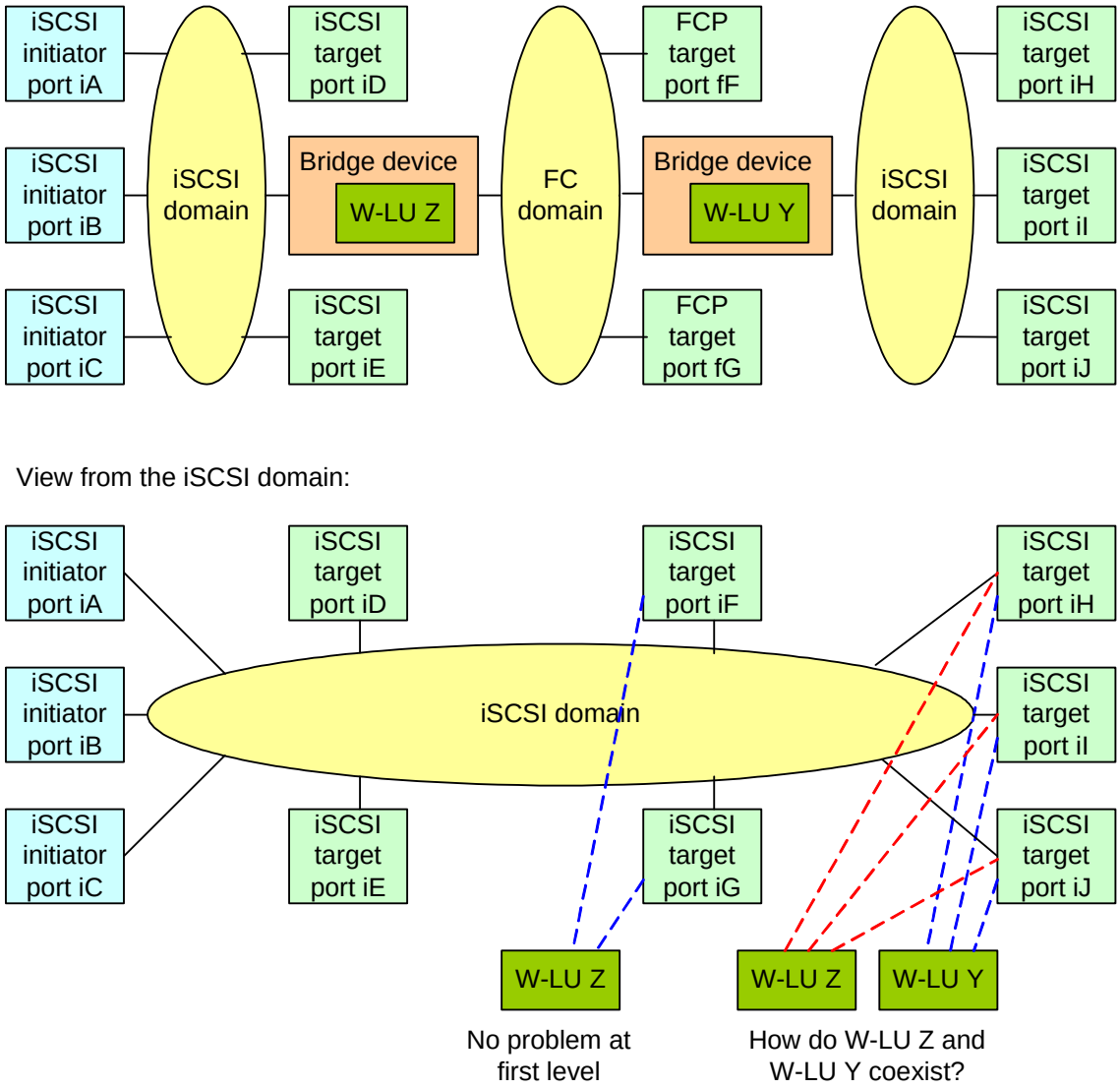

Figure 2 — Nested bridges

The hierarchical addressing format from SAM-3 and SCC-2 can be modified to facilitate this (for 4 levels of bridges).

Table 1 shows logical unit addressing methods defined in SAM-3.

Table 1 — Logical unit addressing methods

Byte\Bit	7	6	5	4	3	2	1	0
Peripheral device addressing								
0	00b		BUS IDENTIFIER					
1	TARGET/LUN							
Flat space addressing								
0	01b							
1	LUN							
Logical unit addressing								
0	10b		TARGET					
1	BUS NUMBER			LUN				
Extended logical unit addressing (used for well-known logical units)								
0	11b		LENGTH		EXTENDED ADDRESS METHOD			
1	EXTENDED ADDRESS METHOD SPECIFIC							

To address a well-known LUN behind a bridge, the 10b format could be replaced with a format that specifies the outgoing relative initiator port identifier instead of unusably small TARGET, BUS NUMBER, and LUN fields. Table 2 shows the proposed type 10b for bridges.

Table 2 — Bridge addressing method

Byte\Bit	7	6	5	4	3	2	1	0
0	10b		(MSB)					
1	RELATIVE INITIATOR PORT							(LSB)

Each bridge shifts off the first level of the LUN address.

With this, to access W-LU Z, an iSCSI initiator port would use the LUN shown in table 3 through target ports iF iG, iH, iI, or iJ.

Table 3 — Accessing W-LU Z

Byte\Bit	7	6	5	4	3	2	1	0
0	11b		LENGTH		EXTENDED ADDRESS METHOD (1h)			
1	EXTENDED ADDRESS METHOD SPECIFIC (W-LUN Z)							(LSB)
2	Unused							
3								
4								
5								
6								
7								

With this, to access W-LU Y in the second bridge, an iSCSI initiator port would use the LUN shown in table 4 through target ports iF, iG, iH, iI, or iJ.

Table 4 — Accessing W-LU Y from iSCSI initiator port

Byte\Bit	7	6	5	4	3	2	1	0
0	10b		(MSB)					
1	RELATIVE INITIATOR PORT (in first bridge)							(LSB)
2	11b		LENGTH		EXTENDED ADDRESS METHOD (1h)			
3	EXTENDED ADDRESS METHOD SPECIFIC (W-LUN Y)							(LSB)
4	Unused							
5								
6								
7								

As the first bridge forwards this LUN, it shifts off the first two bytes, outputting the LUN shown in table 5.

Table 5 — Accessing W-LU Y as seen between first and second bridge

Byte\Bit	7	6	5	4	3	2	1	0
0	11b		LENGTH		EXTENDED ADDRESS METHOD (1h)			
1	EXTENDED ADDRESS METHOD SPECIFIC (W-LUN Y)							(LSB)
2	Unused							
3								
4								
5								
6								
7								

If a third bridge were present with W-LU W, an iSCSI initiator port would use the LUN shown in table 6 through target ports iH, iI, or iJ. Traffic to target ports iF and iG does not go through the second bridge, so the second bridge would not be able to parse the LUN field.

Table 6 — Accessing W-LU W from iSCSI initiator port

Byte\Bit	7	6	5	4	3	2	1	0
0	10b		(MSB)					
1	RELATIVE INITIATOR PORT (in first bridge)							(LSB)
2	10b		(MSB)					
3	RELATIVE INITIATOR PORT (in second bridge)							(LSB)
4	11b		LENGTH		EXTENDED ADDRESS METHOD (1h)			
5	EXTENDED ADDRESS METHOD SPECIFIC (W-LUN W)							(LSB)
6	Unused							
7								

Multiple protocols

A bridge may have more than two transport protocols: for example, it might have iSCSI, FC, and SAS ports, some going into the same domains, some going into different domains. It's probably simplest to assume a 1:1 mapping for now. It can report itself as having three 1:1 bridges in that case (iSCSI to FC, iSCSI to SAS, FC to SAS) perhaps with some duplicated information. This requires using LUN numbers other than a W-LUN for the additional bridges.

1 Problem commands

1.1 Problem commands overview

This clause lists commands that present problems for transparent bridges. Supporting MSC means these issues are all avoided; the bridge can just pass through the data to/from the logical unit unchanged.

1.2 INQUIRY

The Device Identification VPD page (83h) identification descriptors with an ASSOCIATION field set to 1h (i.e., target port) are based on the back side target port. A transparent bridge has to replace these descriptors with ones representing the front side target port.

Identification descriptors with an IDENTIFIER TYPE field set to 4h (relative target port) are based on the back side target port. A transparent bridge preserves this mapping. A LUN mapping bridge changes this mapping; new relative target port numbers might need to be assigned. This affects commands that use the relative target port identifier (see REPORT/SET TARGET PORT GROUPS and PERSISTENT RESERVE IN/OUT).

Identification descriptors with an ASSOCIATION field set to 2h (i.e., target device) are based on the back side target device. A transparent bridge has to replace these descriptors with ones representing the bridge device itself.

1.3 Alias lists (CHANGE/REPORT ALIASES)

Alias entry designators are all protocol specific and need to be translated by a transparent bridge.

1.4 Extended copy (EXTENDED COPY/REPORT COPY RESULTS)

Most target descriptors are protocol specific and need to be translated by a transparent bridge.

1.5 XOR commands (REBUILD/REGENERATE/XDWRITE EXTENDED)

Most target descriptors (shared with EXTENDED COPY) are protocol specific and need to be translated by a transparent bridge.

1.6 Persistent reservations (PERSISTENT RESERVE IN/OUT)

The TransportIDs used by the Specify Initiator Ports feature are protocol specific and need to be translated by a transparent bridge.

Relative target port identifiers returned by the PERSISTENT RESERVE IN command READ FULL STATUS service action (proposed by 03-342) need to be translated by a transparent bridge.

1.7 Access controls (ACCESS CONTROLS IN/OUT)

TransportIDs are protocol specific and need to be translated by a transparent bridge.

AccessIDs, on the other hand, should flow through without problems.

1.8 Asymmetric logical unit access (REPORT/SET TARGET PORT GROUPS)

Relative target port identifiers returned by the PERSISTENT RESERVE IN command READ FULL STATUS service action (proposed by 03-342) need to be translated by a transparent bridge.

1.9 Log pages (LOG SELECT/SENSE)

The Protocol-Specific log page need to be handled by the transparent bridge. (Note: this page includes relative target port identifiers) It is not a good idea to pass through accesses to this page, since the front and back side meanings could be completely different.

1.10 Mode pages (MODE SELECT/SENSE)

The Protocol-Specific Port mode page and Logical Unit mode page need to be handled by a transparent bridge.

It is not a good idea to pass through accesses to these pages, since the front and back side meanings could be completely different.

2 Commands

2.1 REPORT BRIDGE MAPPING temporary overview

Forward (front side to back side) mapping

a) Front side transport protocol identifier
b) Back side transport protocol identifier
c) Front side target port identifiers/LUNs to back side target port identifiers/LUNs mapping
d) Front side initiator port identifiers to back side initiator port identifiers mapping
   A) Identify any limitations on # initiators that can be mapped (e.g. a bridge to parallel SCSI might be limited to 2 initiators). If the limit is 1, then initiator-cognizant commands shall not be used through the bridge.
e) Transparent command support. If the bridge tries to be transparent, what opcodes is the bridge capable of intercepting and replicating on the back side as if they were issued by a native initiator on the back side?

) As this command set is adopted by software, the need for transparent mapping should dwindle. In the interim, this tells a management application what commands application software can safely use without knowing about the bridge.

) Commands that could be intercepted:

   A) INQUIRY
   B) CHANGE/REPORT ALIASES
   C) EXTENDED COPY
   D) REBUILD/REGENERATE/XDWRITE EXTENDED
   E) PERSISTENT RESERVE OUT/IN
   F) ACCESS CONTROL OUT/IN
   G) REPORT/SET TARGET PORT GROUPS
   H) LOG SELECT/SENSE
   I) MODE SELECT/SENSE

Reverse (back side to front side) mapping is also reported.

2.2 REPORT BRIDGE MAPPING command

The REPORT BRIDGE MAPPING command (see table 7) requests information on task management functions (see SAM-3) the addressed logical unit supports.

The REPORT BRIDGE MAPPING command is a service action of the MAINTENANCE IN command. Additional MAINTENANCE IN service actions are defined in SCC-2 and in this standard. The MAINTENANCE

IN service actions defined in SCC-2 apply only to logical units that return a device type of 0Ch or the sccs bit equal to one in their standard INQUIRY data (see SPC-3).

Table 7 — REPORT BRIDGE MAPPING TABLE command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (A3h)							
1	Reserved			SERVICE ACTION (nnh)				
2	Reserved							
5								
6	(MSB)							
9	ALLOCATION LENGTH							(LSB)
10	Reserved							
11	CONTROL							

The ALLOCATION LENGTH field indicates the number of bytes that have been allocated for the returned parameter data. The allocation length shall be at least four. If the allocation length is less than for the command shall be terminated with a CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST, and the additional sense code shall be set to INVALID FIELD IN CDB.

The format of the parameter data returned by the REPORT BRIDGE MAPPING TABLE command is shown in table 8.

Table 8 — REPORT BRIDGE MAPPING TABLE parameter data

Byte\Bit	7	6	5	4	3	2	1	0
0	Reserved				FRONT SIDE PROTOCOL IDENTIFIER			
1	Reserved							
2								
3								
4	Reserved				BACK SIDE PROTOCOL IDENTIFIER			
5	Reserved							
6								
7								
8	EXTENDED COPY	ACCESS CONTROL	PERSISTENT RESERVE	TARGET PORT GROUPS	ALIAS	MODE	LOG	INQUIRY
9	Reserved							XOR
10	Reserved							
11	NUMBER OF FORWARD MAPPING TABLE ENTRIES							
	Forward mapping table entries							
12	Forward mapping table entry (first)							
	...							
	Forward mapping table entry (first)							
m								
m+1	NUMBER OF BACKWARD MAPPING TABLE ENTRIES							
	Backward mapping table entries							
m+2	Backward mapping table entry (first)							
	...							
	Backward mapping table entry (first)							
n								

The FRONT SIDE PROTOCOL IDENTIFIER field contains the protocol identifier (see SPC-3) of the front side initiator port(s) and target port(s).

The BACK SIDE PROTOCOL IDENTIFIER field contains the protocol identifier (see SPC-3) of the front side initiator port(s) and target port(s).

An INQUIRY bit set to one means the bridge intercepts the INQUIRY command. An INQUIRY bit set to one means the bridge does not intercept the INQUIRY command.

[repeat for all the command intercept bits]

The NUMBER OF FORWARD MAPPING TABLE ENTRIES field indicates how many forward mapping table entries follow.

The NUMBER OF BACKWARD MAPPING TABLE ENTRIES field indicates how many forward mapping table entries follow.

The forward mapping table format is defined in table 9.

Table 9 — Forward mapping table entry

Byte\Bit	7	6	5	4	3	2	1	0
0	(MSB)	ENTRY LENGTH (n - 1)						
1								(LSB)
2	(MSB)	FRONT SIDE TARGET DESCRIPTOR LENGTH (x - 3)						
3								(LSB)
4	FRONT SIDE TARGET DESCRIPTOR							
x								
x + 1	(MSB)	BACK SIDE INITIATOR TRANSPORTID LENGTH (y - x)						
x + 2								(LSB)
x + 3	BACK SIDE INITIATOR TRANSPORTID							
y								
y + 1	(MSB)	BACK SIDE TARGET DESCRIPTOR LENGTH (n - y)						
y + 2								(LSB)
y + 3	BACK SIDE TARGET DESCRIPTOR							
n								

The ENTRY LENGTH field indicates the number of bytes that that follow in the forward mapping table entry.

The FRONT SIDE TARGET DESCRIPTOR LENGTH field indicates the number of bytes in the FRONT SIDE TARGET DESCRIPTOR field.

The FRONT SIDE TARGET DESCRIPTOR field contains an EXTENDED COPY target descriptor (see SPC-3) identifying a target port and one or more logical units. The LU IDENTIFIER field of the target descriptor shall be set to FFFFFFFF FFFFFFFFh if all logical units behind the target port are being mapped.

Editor's Note 1: Instead of the front side target descriptor, the relative target port and LUN field could be specified. The application can convert that into a full fledged target identifier using the All Target Ports VPD page proposed in 03-344. This would shorten the data structure and avoid confusion with the RELATIVE INITIATOR PORT field that is being added to the target descriptor by 03-354.

The BACK SIDE INITIATOR TRANSPORTID LENGTH field indicates the number of bytes in the BACK SIDE INITIATOR TRANSPORTID field.

The BACK SIDE INITIATOR TRANSPORTID field contains a TransportID (see SPC-3) identifying the initiator port used for accessing the target port.

Editor's Note 2: Instead of the back side initiator TransportID, the relative initiator port could be specified. The application can convert that into a TransportID using the All Initiator Ports VPD page

proposed in 03-353.

The BACK SIDE TARGET DESCRIPTOR LENGTH field indicates the number of bytes in the BACK SIDE TARGET DESCRIPTOR field.

The BACK SIDE TARGET DESCRIPTOR field contains an EXTENDED COPY target descriptor (see SPC-3) identifying a target port and one or more logical units that a front side initiator port accesses as the front size target port indicated by the FRONT SIDE TARGET DESCRIPTOR field with the bridge initiator port indicated by the BACK SIDE INITIATOR TRANSPORTID field. The LU IDENTIFIER field of the target descriptor shall be set to FFFFFFFF FFFFFFFFh if all logical units behind the target port are being mapped.

The backward mapping table format is defined in table 10.

Table 10 — Backward mapping table entry

Byte\Bit	7	6	5	4	3	2	1	0
0	(MSB)	ENTRY LENGTH (n - 1)						
1								(LSB)
2	(MSB)	BACK SIDE TARGET DESCRIPTOR LENGTH (x - 3)						
3								(LSB)
4		BACK SIDE TARGET DESCRIPTOR						
x								
x + 1	(MSB)	FRONT SIDE INITIATOR TRANSPORTID LENGTH (y - x)						
x + 2								(LSB)
x + 3		FRONT SIDE INITIATOR TRANSPORTID						
y								
y + 1	(MSB)	FRONT SIDE TARGET DESCRIPTOR LENGTH (n - y)						
y + 2								(LSB)
y + 3		FRONT SIDE TARGET DESCRIPTOR						
n								

The ENTRY LENGTH field indicates the number of bytes that that follow in the backward mapping table entry.

The BACK SIDE TARGET DESCRIPTOR LENGTH field indicates the number of bytes in the BACK SIDE TARGET DESCRIPTOR field.

The BACK SIDE TARGET DESCRIPTOR field contains an EXTENDED COPY target descriptor (see SPC-3) identifying a target port and one or more logical units. The LU IDENTIFIER field of the target descriptor shall be set to FFFFFFFF FFFFFFFFh if all logical units behind the target port are being mapped.

Editor's Note 3: Instead of the back side target descriptor, the relative target port and LUN field could be specified. The application can convert that into a full fledged target identifier using the All Target Ports VPD page proposed in 03-344. This would shorten the data structure and avoid confusion with the RELATIVE INITIATOR PORT field that is being added to the target descriptor by 03-354.

The FRONT SIDE INITIATOR TRANSPORTID field indicates the number of bytes in the FRONT SIDE TARGET DESCRIPTOR field.

The FRONT SIDE INITIATOR TRANSPORTID field contains a TransportID (see SPC-3) identifying the initiator port used for accessing the target port.

Editor's Note 4: Instead of the front side initiator TransportID, the relative initiator port could be specified. The application can convert that into a TransportID using the All Initiator Ports VPD page proposed in 03-353.

The FRONT SIDE TARGET DESCRIPTOR LENGTH field indicates the number of bytes in the FRONT SIDE TARGET DESCRIPTOR field.

The FRONT SIDE TARGET DESCRIPTOR field contains an EXTENDED COPY target descriptor (see SPC-3) identifying a target port and one or more logical units that a front side initiator port accesses as the front size target port indicated by the FRONT SIDE TARGET DESCRIPTOR field with the bridge initiator port indicated by the BACK SIDE INITIATOR TRANSPORTID field. The LU IDENTIFIER field of the target descriptor shall be set to FFFFFFFF FFFFFFFFh if all logical units behind the target port are being mapped.

2.3 WAIT FOR BRIDGE MAPPING CHANGE command

The WAIT FOR BRIDGE MAPPING command (see table 11) waits for a bridge mapping to change before completing. After this command completes, an application client should send a REPORT BRIDGE MAPPING command to determine what changed.

The WAIT FOR BRIDGE MAPPING CHANGE command is a service action of the MAINTENANCE IN command. Additional MAINTENANCE IN service actions are defined in SCC-2 and in this standard. The MAINTENANCE IN service actions defined in SCC-2 apply only to logical units that return a device type of 0Ch or the sccs bit equal to one in their standard INQUIRY data (see SPC-3).

Table 11 — WAIT FOR BRIDGE MAPPING TABLE command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (A3h)							
1	Reserved			SERVICE ACTION (nnh)				
2	Reserved							
9								
10	Reserved							
11	CONTROL							