

Document: T10/00-279r0
To: T10 Committee Membership
From: Edward A. Gardner, Ophidian Designs
Subject: Large SCSI Device Identifiers

Date: July 8, 2000

The current SCSI standards specify that every SCSI device shall have a 64-bit device identifier. Quoting from SAM-2 revision 13, page 24:

An Initiator Identifier is a field containing up to 64 bits that is a SCSI device identifier for the initiator device.

A Target Identifier is a field containing up to 64 bits that is a SCSI device identifier for the target device.

Unfortunately, 64-bits is no longer sufficient to identify a device with forthcoming protocols. For example, SVP will need to identify an initiator or target by a VI address. Many VI implementations construct a VI address from an IPv6 address plus a discriminator. Since an IPv6 address is 128 bits, this cannot fit within a 64-bit field. For another example consider the various proposals for SCSI over IP (Internet Protocol). At least one, iSCSI, is proposing to identify devices by URL strings.

There are two approaches to dealing with this problem. One is to have each protocol that encounters this problem define a protocol specific device identifier mapping. For example, SVP would define an SVP specific 64-bit to VI address identifier mapping, with a protocol specific way for initiators to load and maintain the mapping table in targets. While that could be done, I believe it is fundamentally a kludge and will lead to long-term support and compatibility problems.

The other approach is to alter the relevant SCSI standards to allow much larger device identifiers. This document sketches how that might be done. If the committee agrees that this is the proper approach, I will prepare detailed proposals for the affected standards for discussion at this fall's T10 meetings.

So far as I am aware, device identifier size only affects third-party operations, those where one target is asked to become a temporary initiator to access another target. The two important third-party operations are EXTENDED COPY and the third-party disk XOR operations. My impression is that other third-party operations (e.g. third-party RESERVATION) are obsolete and need not be supported. If anyone disagrees with these assumptions, I trust they will let us know.

With these assumptions, the affected documents are SAM-2, SPC-3 and SBC-2. The changes I plan to propose to each are summarized below.

SAM-2

The existing statements about the size of device identifiers would be removed. If SAM-2 says anything at all about device identifier size, it should say that it is protocol specific.

SPC-3

The EXTENDED COPY command references a lengthy parameter list that controls its operation. The parameter list contains a header, a list of target descriptors, a list of segment descriptors and an inline data area.

Target descriptors are fixed length, 32-byte constructs. They include a 24-byte field available to identify the target and LUN. Various formats of target descriptors are defined that include Fibre Channel and parallel SCSI device identifiers.

Segment descriptors define a data transfer to be performed by the EXTENDED COPY command. They identify source and destination target devices with offsets into the list of target descriptors. Various formats define various kinds of data transfer operations. One format copies data from the inline data area to the destination device, presumably for writing labels and other control information. The inline data to be copied is identified by a 4-byte offset and a 4-byte length.

I plan to propose one or more new target descriptor formats. The actual target identifier would be stored in the inline data area. The new target descriptor format would contain a LUN (8 bytes), offset to the target identifier (4 bytes), length of the target identifier (2 or 4 bytes) and a target identifier format code (2 bytes). Target identifier format codes would include values for unspecified, FC-VI, Infiniband, etc. The format codes serve much the same purpose as the current multiple formats of Fibre Channel target descriptors, one for port identifier, one for port WWN, one for both, etc.

I am also tempted to allow the LUN's device identification VPD data to be specified (e.g. the LUN's WWN) within the inline data area, analogous to the current Identification descriptor target descriptor format. I welcome comments on whether this would be useful.

SBC-2

The third-party XOR commands identify the secondary target using a one byte SECONDARY ADDRESS field. For parallel SCSI this is the target identifier. For Fibre Channel it specifies the low byte of the second target's port identifier; the other bytes of the port identifier are the same as the primary target's.

The third-party XOR commands also contain a TABLE ADDRESS flag. When that flag is zero, the commands operate as described above. When that flag is one, SBC / SBC-2 specify:

A TABLE ADDRESS bit of one indicates that the SECONDARY ADDRESS field contains a pointer to a look up table of ANSI X3.270 SAM compliant target identifiers. The look up table is reserved for future definition.

I plan to propose that we define the format of the lookup table for SBC-2. I plan to propose a table format that is a strict subset of that used by EXTENDED COPY. The header, target descriptors and inline data area would be identical. Segment descriptors would be prohibited. I believe that all the target descriptor formats defined by EXTENDED COPY should be supported.

I would like to have SBC-2 cross-reference SPC-3 for the table format, rather than redefining it within SBC-2. Is that legitimate? Does it require any re-structuring of either standard?

I believe the table should be volatile, reset on power-up or anything that resets the LUN. There should be exactly one copy per LUN, shared by all initiators. The maximum table length is close to 256*32 or 8192 bytes plus the inline data, perhaps about 64Kbytes total. That's too large to fit easily into our mode page structure. We could segment the table, but I don't like that, I think it should be loaded by a single command, The other choices are a new command or to define one of the reserved bytes in MODE SELECT to indicate the type of mode information provided. I have no preference between these last two choices. I would also welcome other suggestions.