To: Membership of X3T10

From: Ralph O. Weber
Digital Equipment Corporation

Date: May 30, 1994

Subject: Reserve & Release in SCSI-3 Primary Commands

At the March working group, I received guidance on building a RESERVE and RELEASE command structure that works with the 64-bit SCSI device identifiers and logical unit identifiers defined by the SAM. This document contains proposed command definitions for RELEASE(6), RELEASE(10), RESERVE(6), and RESERVE(10). The definitions are base on SCSI-2 revision 10k and George Penokie's 16/32 bit P/Q and L cables document.

Also, I have tried to combine RESERVE/RELEASE from the disk world with RESERVE UNIT/RELEASE UNIT from the tape world. Since the SPC covers all device models, having a single command definition for disks and tapes seems like the desired approach.

Those of you reading this document from the SCSI BBS or on the SCSI Reflector will notice phrases such as, ~~<<this is a test>>~~. The double angle brackets delimit text that appeared in SCSI-2 that I think can be removed in SCSI-3. The hardcopy version contains a strikeout line through the text.

The following text goes in the device models clause.

x.x Reservations

The access enabled or access disabled condition determines when an application client may store or retrieve user data on all of the medium. For random access devices, access to the medium may be restricted to specified parts of the medium for read operations, write operations, or both. These attributes may be controlled by an external mechanism or by the RESERVE and RELEASE commands (see ?.?.?).

The RESERVE and RELEASE commands define how different types of restricted access may be achieved, and to whom the access is restricted. This subclause describes the interaction of the application clients on the initiator that requested the reservation with application clients on other initiators.

An application client uses reservations to gain a level of exclusivity in access to all or part of the medium for itself or an applicaton client on another initiator. Because a device server cannot differentiate between different application clients on an initiator, all application clients on the same initiator have the same access. It is expected that the reservation will be retained until released. The SCSI device must ensure that the initiator with the reservation is able to access the reserved media within the operating parameters established by the application client on that initiator.

The following paragraphs explain~~<<, on a command by command basis,>>~~ the appropriate device server response when a reservation exists. Unless otherwise noted, the appropriate response to an application client that issues a command to a SCSI device that is reserved to another initiator is RESERVATION CONFLICT status.

Unless specific reservation conflict rules are stated for a given command the following general rules shall apply. A reservation conflict shall occur when the entire unit is reserved and the device server receives a command from an application client on an initiator other than the one holding the reservation. Commands that affect overall device status shall generate a reservation conflict, if the device server receives them while an extent reservation is present. ALL other commands that request read or write operations shall be evaluated for reservation conflict as described in the RESERVE command (see clause ?.?.?).

The individual command standards (SBC, SSC, SGC, etc.) may define other specific commands that are affected in specific ways by reservations from another initiator. The INQUIRY and REQUEST SENSE commands shall not be affected by any kind of reservation. The LOG SELECT, LOG SENSE, MODE SENSE, TEST UNIT READY, READ BUFFER, and WRITE BUFFER commands shall not be affected by extent reservations.

The MODE SELECT command shall be dealt with as follows. If an initiator has an extent reservation on a SCSI device, and an application client on another initiator sends a MODE SELECT, a reservation conflict shall occur if the MODE SELECT affects the manner in which access to an extent reserved by the first initiator is performed. If the MODE SELECT does not affect access to the extent, or mode parameters are saved for each initiator, then a conflict shall not occur.

The CHANGE DEFINITION command is dealt with as follows. If any initiator has an extent reservation on a SCSI device, no other initiator may affect the operating definition of that initiator holding the reservation by use of this command. If the SCSI device allows different operating definitions for each initiator, then there is no conflict; otherwise, a reservation conflict shall occur.

The COMPARE, COPY, and COPY AND VERIFY commands are evaluated for reservation conflict as if they were normal write and read operations even when a SCSI device is requested to copy to or from itself. For example, if a COPY is issued to logical unit 0 that requests the SCSI device to copy from logical unit 0 to logical unit 1, access to logical unit 1 must also be evaluated for conflict. COPY commands shall be terminated with CHECK CONDITION status and the sense key shall be set to DATA PROTECT if any part of the copy operation is prohibited by an active extent reservation.

The SEND DIAGNOSTIC, RECEIVE DIAGNOSTIC RESULTS commands conflict with an extent reservation only if they affect access to the extent (as with MODE SELECT).

When a system is integrated with more than one initiator, there must be agreement between the initiators as to how media is reserved and released during operations, otherwise, an initiator may be locked out of access to a target in the middle of an operation. For example, initiator 'A' has a write operation in progress to a direct-access device which has data stored in cache memory. Then, initiator 'B' issues a RESERVE command to the direct-access device. As a result, initiator 'A' is locked out of issuing a SYNCHRONIZE CACHE command to ensure the integrity of the data. To prevent this from happening, initiator 'A' should issue a RESERVE prior to the write command.

The following text defines the RELEASE(6), RELEASE(10), RESERVE(6), and RESERVE(10) commands.

x.x RELEASE(6) command

The RELEASE(6) command (see table a1) is used to release a previously reserved logical unit, or, if the extent release option is implemented, to release previously reserved extents within a logical unit.

Table a1 - RELEASE(6) command

Bit Byte	7	6	5	4	3	2	1	0
0	Operation code (17h)							
1	Reserved			3rdPty	Third party device ID			Extent
2	Reservation identification							
3	Reserved							
4	Reserved							
5	Control							

The RESERVE and RELEASE commands provide the basic mechanism for contention resolution in multiple-initiator systems. A reservation may only be released by an application client on the initiator that made it. It is not an error for an application client to attempt to release a reservation that is not currently valid, or is held by an application client on another initiator. In this case, the device server shall return GOOD status without altering any other reservation.

x.x.x Logical unit release (Mandatory)

If the extent bit is zero, this command shall cause the device server to terminate all non-third-party logical unit and extent reservations that are active from the initiator to the specified logical unit. The reservation ID field in the command descriptor block shall be ignored by the device server.

x.x.x Extent release (Optional)

If the extent bit is one and the extent release option is not implemented, then the RELEASE command shall be terminated with CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST. This option shall be implemented if the extent reservation option (see ?.?.?) is implemented.

If the extent bit is one and the extent release option is implemented, this command shall cause any reservation from the requesting initiator with a matching reservation identification to be terminated. Other reservations from the requesting initiator shall remain in effect.

x.x.x Third-party release (Mandatory)

Third-party release allows an application client to release a logical unit or extents within a logical unit that were previously reserved using third-party reservation (see ?.?.?). Third-party release shall be implemented. It is intended for use in multiple-initiator systems that use the COPY command.

If the third-party (3rdPty) bit is zero, then a third-party release is not requested. If the 3rdPty bit is one then the device server shall release the specified logical unit or extents, but only if the reservation was made using a third-party reservation by an application client on the initiator that is requesting the release for the same SCSI device as specified in the third-party device ID field.

In the RELEASE(6) command, the Third party device ID field is restricted to three bits (values 0 to 7). If larger Third party device ID values are required, the RELEASE(10) command must be used.

If the 3rdPty bit is one the device server shall not modify the mode parameters for commands received from the third-party device even if the target implements the transfer of mode parameters with a third-party RESERVE command.

> NOTE n1 If a target implements independent storage of mode parameters for each initiator, a third-party RESERVE command copies the current mode parameters for the initiator that sent the RESERVE command to the current mode parameters for the initiator specified as the third-party device (usually a copy master device). A unit attention condition notifies the third-party of the changed mode parameters due to the reservation. A successful third-party RELEASE command does not return the third-party devices' current mode parameters back to their previous values. The third-party device can issue MODE SENSE and MODE SELECT commands to query and modify the mode parameters.

x.x RELEASE(10)

The RELEASE(10) command (see table a2) is used to release a previously reserved logical unit, or, if the extent release option is implemented, to release previously reserved extents within a logical unit.

Table a2 - RELEASE(10) command

Bit Byte	7	6	5	4	3	2	1	0
0	Operation code (57h)							
1	Reserved			3rdPty	Reserved			Extent
2	Reservation identification							
3	Third party device ID							
4	Reserved							LongID
5	Reserved							
6	Reserved							
7	(MSB) Parameter list length							
8								(LSB)
9	Control							

If the Third party device ID value associated with the reservation release is smaller than 255, the LongID bit may be zero and the ID value sent in the CDB. If the Third party device ID is greater than 255, the LongID bit shall be one. If the LongID bit is one, the Parameter list length shall be eight, and the parameter list shall have the format shown in table a3. If the LongID bit is one and the Parameter list length is not eight, the device server shall return a CHECK CONDITION status with a sense key of ILLEGAL REQUEST. If the LongID bit is zero, the Parameter list length field shall be reserved.

Table a3 - RELEASE(10) parameter list

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)			Third party device ID				
7								(LSB)

In all other respects and for all other fields, the RELEASE(10) command shall function exactly like the RELEASE(6) command.

x.x RESERVE(6) command

The RESERVE(6) command (see table a4) is used to reserve a logical unit or, if the extent reservation option is implemented, extents within a logical unit.

Table a4 - RESERVE(6) command

Bit Byte	7	6	5	4	3	2	1	0
0	Operation code (16h)							
1	Reserved			3rdPty	Third party device ID			Extent
2	Reservation identification							
3	(MSB)			Extent list length				
4								(LSB)
5	Control							

The RESERVE and RELEASE commands provide the basic mechanism for contention resolution in multiple-initiator systems. The third-party reservation allows logical units or extents to be reserved for another specified SCSI device.

If the RESERVE(6) command is implmented, then the RELEASE(6) also shall be implemented.

x.x.x Logical unit reservation (Mandatory)

If the extent bit is zero, this command shall request that the entire logical unit be reserved for the exclusive use of the initiator until the reservation is superseded by another valid RESERVE command from the same initiator ~~<<that made the reservation >>~~or until released by a RELEASE command from the same initiator that made the reservation, by a BUS DEVICE RESET message from any initiator, by a hard RESET condition, or by a power on cycle. A logical unit reservation shall not be granted if the logical unit or any extent is reserved by another initiator. It shall be permissible for an initiator to reserve a logical unit that is currently reserved by that initiator. If the extent bit is zero, the reservation identification and the extent list length shall be ignored.

If the logical unit, or any extent within the logical unit is reserved for another initiator, the device server shall return RESERVATION CONFLICT status.

If, after honouring the reservation, an application client on any other initiator attempts to perform any command on the reserved logical unit other than an INQUIRY, REQUEST SENSE,~~<< PREVENT ALLOW MEDIUM REMOVAL (...),>>~~ or a RELEASE command the command shall be rejected with RESERVATION CONFLICT status. The individual command standards (SBC, SSC, SGC, etc.) may define other specific commands that are permissable while there is a reservation from other initiator.

x.x.x Extent reservation (Optional)

The reservation identification field provides a means for an application client to identify each extent reservation. This allows an application client in a multiple tasking environment, to have multiple reservations outstanding. The reservation identification is used in the RELEASE command to specify which reservation is to be released. It is also used in superseding RESERVE commands to specify which reservation is to be superseded.

If the extent reservation option is implemented, then the extent release option (see ?.?.?) shall also be implemented. These options permit multiple extents within the logical unit to be reserved, each with a separate reservation type.

If the extent bit is one, and the extent reservation option is implemented, then the device server shall process the reservation request as follows:

a) The extent list shall be checked for the number of extents in the reservation request. If the extent list length is zero, no current reservations shall be changed, no new reservations shall be created, and this condition shall not be treated as an error. If the extent list contains more extents than are supported on the logical unit, the command shall be terminated with CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST. If the extent list contains more extents than are currently available on the logical unit, then the device server shall return a RESERVATION CONFLICT status.
b) The extent list shall be checked for valid extent logical block addresses. If any logical block address is invalid for this logical unit, the command shall be terminated with CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST. The extent list shall be checked for invalid extent overlaps (as defined by reservation type) with other extent descriptors in the extent list and if invalid overlaps are found, the command shall be terminated with CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST.
c) If the requested reservation does not conflict with an existing reservation, the extents specified shall be reserved until superseded by another valid RESERVE command from an application client on the initiator that made the reservation or until released by a RELEASE command from the same initiator, by a BUS DEVICE RESET message from any initiator, or by a hard RESET condition. If either of the last two conditions occur, a unit attention condition shall be generated~~<< the next command from each initiator shall be terminated with CHECK CONDITION status and the sense key shall be set to UNIT ATTENTION>>~~.
d) If the reservation request conflicts with an existing reservation, then the device server shall return a RESERVATION CONFLICT status.

If the extent bit is one, and the extent reservation option is not implemented, then the RESERVE command shall be rejected with CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST.

The size of the extent list shall be defined by the extent list length field. The extent list shall consist of zero or more descriptors as shown in table a5. Each extent descriptor defines an extent beginning at the specified logical block address for the specified number of blocks. If the number of blocks is zero, the extent shall begin at the specified logical block address and continue through the last logical block address on the logical unit.

Table a5 - Data format of extent descriptors

Bit Byte	7	6	5	4	3	2	1	0
0	Reserved					RelAdr	Reservation type	
1	(MSB)							
	Number of blocks							
3								(LSB)
4	(MSB)							
	Logical block address							
7								(LSB)

The reservation type field shall define the type of reservation in effect for the extent. The types of reservation are defined in table a6.

Table a6 - Reservation types

Reservation type	Description
00b	Read shared
01b	Write exclusive
10b	Read exclusive
11b	Exclusive access

a) **Read exclusive.** While this reservation is active, no other initiator shall be permitted read operations to the indicated extent. This reservation shall not inhibit write operations from any initiator or conflict with a write exclusive reservation; however, read exclusive, exclusive access, and read shared reservations that overlap this extent shall conflict with this reservation.

b) **Write exclusive.** While this reservation is active, no other initiator shall be permitted write operations to the indicated extent. This reservation shall not inhibit read operations from any initiator or conflict with a read exclusive reservation from any initiator. This reservation shall conflict with write exclusive, exclusive access, and read shared reservations that overlap this extent.

c) **Exclusive access.** While this reservation is active, no other initiator shall be permitted any access to the indicated extent. All reservation types that overlap this extent shall conflict with this reservation.

d) **Read shared.** While this reservation is active, no write operations shall be permitted by any initiator to the indicated extent. This reservation shall not inhibit read operations from any initiator or conflict with a read shared reservation. Read exclusive, write exclusive, and exclusive access reservations that overlap with this extent shall conflict with this reservation.

If the relative address bit is one, the logical block address in the extent descriptor shall be treated as a two's complement displacement. This displacement shall be added to the logical block address last accessed on the logical unit to form the logical block address for this extent. This feature is only available when linking commands and requires that a previous command in the linked group has accessed a logical block on the logical unit; if not, the RESERVE command shall be terminated with CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST.

If an application client attempts a command to a logical block that has been reserved and that access is prohibited by the reservation, the command shall not be performed and the command shall be terminated with a RESERVATION CONFLICT status. If a reservation conflict precludes any part of the command, none of the command shall be performed. Clause (?.?) and the various command standards (SBC, SSC, SGC, etc.) define additional rules for interactions between extent reservations and specific commands.

x.x.x Third-party reservation (Mandatory)

The third-party reservation for the RESERVE command allows an application client to reserve a logical unit or extents within a logical unit for another SCSI device. This is intended for use in multiple-initiator systems that use the COPY command.

If the third-party (3rdPty) bit is zero, then a third-party reservation is not requested. If the 3rdPty bit is one then the device server shall reserve the specified logical unit or extents for the SCSI device specified in the third-party device ID field. The device server shall preserve the reservation until it is superseded by another valid RESERVE command from the initiator that made the reservation or until it is released by the same initiator, by a BUS DEVICE reset message from any initiator, or a hard reset condition. The device server shall ignore any attempt to release the reservation made by any other initiator.

In the RESERVE(6) command, the Third party device ID field is restricted to three bits (values 0 to 7). If larger Third party device ID values are required, the RESERVE(10) command must be used.

If independent sets of mode parameters are implemented, a third party reservation shall cause the device server to transfer the set of mode parameters in effect for the application client that sent the RESERVE command to the mode parameters used for commands from the third party device. Any subsequent command issued by the third-party device shall be executed according to the mode parameters in effect for the application client that sent the RESERVE command.

NOTE n2 This transfer of the mode parameters is applicable to device servers that store mode information independently for different initiators. This mechanism allows an application client to set the mode parameters of a target for the use of a copy master (i.e. the third-party device). The third-party copy master may subsequently issue a MODE SELECT command to modify the mode parameters.

x.x.x Superseding reservations (Mandatory)

Implementation of superseding reservations is mandatory. An application client that holds a current reservation (unit or extent) may modify that reservation by issuing another RESERVE command (unit or extent) to the same logical unit. The superseding RESERVE command shall release the previous reservation state (unit or extent) when the new reservation request is granted. If the superseding reservation is for an extent reservation and the current reservation is also an extent reservation, the current extent reservation identification value is used for the superseding reservation. The current reservation shall not be modified if the superseding reservation request cannot be granted. If the superseding reservation cannot be granted because of conflicts with a previous reservation (other than the reservation being superseded), then the device server shall return RESERVATION CONFLICT status.

NOTE n3 Superseding reservations allow the SCSI device ID to be changed on a reservation using the third-party reservation option. This capability is necessary for certain situations when using COMPARE, COPY, and COPY AND VERIFY commands.

x.x RESERVE(10)

The RESERVE(10) command (see table a7) is used to reserve a logical unit or, if the extent reservation option is implemented, extents within a logical unit.

Table a2 - RESERVE(10) command

Bit Byte	7	6	5	4	3	2	1	0
0	Operation code (56h)							
1	Reserved			3rdPty	Reserved			Extent
2	Reservation identification							
3	Third party device ID							
4	Reserved							LongID
5	Reserved							
6	Reserved							
7	(MSB)	Parameter list length						
8								(LSB)
9	Control							

The RESERVE(10) and RELEASE(10) commands provide the basic mechanism for contention resolution in multiple-initiator systems. The third-party reservation allows logical units or extents to be reserved for another specified SCSI device.

If the RESERVE(10) command is implmented, then the RELEASE(10) also shall be implemented.

If the Third party device ID value associated with the reservation release is smaller than 255, the LongID bit may be zero and the ID value sent in the CDB. If the Third party device ID is greater than 255, the LongID bit shall be one. If the LongID bit is one, the Parameter list length shall be at least eight. If the LongID bit is one and the Parameter list length is less than eight, the device server shall return a CHECK CONDITION status with a sense key of ILLEGAL REQUEST.

If both the LongID and Extent bits are one, then the parameter list shall have the format shown in table a8 and the extent list length shall be the Parameter list length minus eight.

Table a8 - RESERVE(10) ID & extents parameter list

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
	Third party device ID							
7								(LSB)
8								
	Extent descriptors (see table a5)							
n								

If the LongID bit is one and the Extent bit is zero, the Parameter list length shall be eight, and the parameter list shall have the format shown in table a9. If the LongID bit is one and the Extent bit is zero and the Parameter list length is not eight, the device server shall return a CHECK CONDITION status with a sense key of ILLEGAL REQUEST.

Table a9 - RESERVE(10) ID only parameter list

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
	Third party device ID							
7								(LSB)

If the LongID bit is zero, the Parameter list shall be processed as an extent list (see ?.?).

In all other respects and for all other fields, the RESERVE(10) command shall function exactly like the RESERVE(6) command.