

Date: August 27, 2003

To: T10 Committee (SCSI)

From: Roger Cummings (VERITAS)

Subject: "Only If Reserved" Proposal

This proposal has arisen from recent research into Persistent Reservations, but it is a separate proposal which stands alone from that work. The feature proposed here will work with (Original) Reservations, or any type of Persistent Reservations, and will directly address some significant problems encountered by designers of backup and restore applications. Adoption of this proposal will allow simplification of existing code, and provide much-improved resistance against a particularly destructive class of data corruption errors that have been encountered in a number of situations in the field.

## BACKGROUND

As stated in 03-231r0, which gives an overview of the latest investigations on the use of Persistent Reservations with serial access devices, it has been discovered that there are certain operating systems, drivers and applications that are profligate with the use of reserves and releases, that is, they issue many of each type during the execution of specific code paths.

This situation occurs because there is no way to test that an (original) reservation is still in existence for an I\_T nexus. Issuing a RESERVE command returns the same status if a reservation to the issuing I\_T nexus existed before the command was received or not. Similarly, issuing a RELEASE command returns the same status if a reservation to the issuing I\_T nexus existed before the command was received or not.

Designers of backup and restore applications are very concerned to avoid to avoid writing to unreserved serial access devices, because they can be subject to later and almost undetectable data corruption resulting from accesses from other Initiators (either servers or data copy engines). The combination of this concern and the lack of a way of testing a reservation results in the profligate use of reserves described above.

However the provision of a test facility would not by itself solve this problem, because by definition there will always be a time interval between the completion of the test and the issuance of a data transfer command. To avoid this "hole" what is needed is a way to ensure that a command will only be executed by the device server if a reservation exists for the I\_T nexus under which the command is issued. If no such reservation exists, the command would not be performed, and the device server would return a new status to indicate that the device server is not reserved.

## PROPOSAL

The document proposes the definition of a new bit for the Control mode page, called OIR. The positioning of this bit within the page is at the pleasure of the CAP group, with bit 7 of byte 5 being the author's suggestion. The definition of this bit has been extended from the background given above to also cover persistent reservations

The additional text for section 7.4.6 is proposed as follows, with the exact positioning of course determined by the bit selection as above:

*A only if reserved (OIR) bit set to one specifies that the device server shall only perform a command if a reservation or persistent reservation exists which allows access to the I\_T nexus from which the command was received. When OIR is one and a command is received from an I\_T nexus for which no a reservation or persistent reservation exists, the command shall not be performed and the device server shall terminate the command with CHECK CONDITION status, and shall set the sense key to NOT READY and the additional sense code to NOT RESERVED. For a list of commands affected by the OIR bit, see the tables that define the commands allowed in the presence of various reservations in this standard (Table 31) and in the command standard (see 3.1.17) for the specific device type. Any command which has "Conflict" in any column is affected by OIR bit.*

This document also proposes the definition of an additional ASCQ in Table 28 and Annex C to indicate NOT RESERVED. The value used is again at the pleasure of the CAP group.