SCSI Device Memory Export Protocol

Version 0.9.8

August 21, 2000

Andrew Barry, Kenneth Preslan, Matthew O'Keefe
Sistina Software
1313 5th St. S.E. Suite 111 Minneapolis, MN 55414
+1-612-379-3951, {barry, kpreslan, okeefe}@sistina.com
http://www.sistina.com

Gerry Johnsen - Ciprico Inc.

James Wayda - Dot Hill Systems Corp.

Burn Alting - Comptex Pty. Ltd.

Contents

Abstract

Cluster File Systems are a versatile, high performance form of distributed file access. Unlike traditional distributed file systems, Cluster File Systems can utilize Storage Area Networks to provide local file system performance to dozens or hundreds of computers. Symmetric Shared Disks File Systems offer the added benefit of increased scalability and reliability because no single machine is responsible for managing the file system. For these File Systems to work, they require a global lock space which is accessible to all clients. Storage devices are a good place to put these locks, both because there has already been a lot of effort made to make them reliable, and because putting the lock space and the data on the same device reduces the number of fail-able components in the CFS. The following is a proposal for the implementation of the SCSI Memory Export commands. It can be used as part of a locking protocol to provide a global lock space for a CFS. This paper details how SCSI Memory Export behaves, how it is used as a mutual exclusion primitive, and provides an example of its implementation.

1 SCSI Memory Export and SCSI Device Locks

The proposal for SCSI Memory Export evolved from various versions of an earlier proposal, SCSI Device Locks. Over several years, the proposed SCSI Device Locks had grown into a complicated, sophisticated locking mechanism for cluster mutual exclusion. Though very powerful as a mutual exclusion lock, and as a detector or cluster failure, device locks had grown very difficult to implement. To ease the difficulty of implementation and to encourage more vendors to implement a cluster locking method, we put forward this proposal as a replacement to the proposed SCSI Device Locks.

SCSI Memory Export is a generic space of memory buffers on SCSI storage devices. These buffers are readable and writable by clients using a Load Locked/Store Conditional mechanism enforced by sequence numbers. The buffers are identified with 72-bit Logical Buffer Identifier, which is mapped to a physical memory buffer on the SCSI device. These buffers form a sparse space where there are many more possible Logic Buffer Identifiers than there are physical memory buffers. But because only a small percentage of Logical Buffer Numbers are active at a given time, a one-to-one mapping can be maintained.

The Memory Export specification allows SCSI device vendors to implement a relatively simple, generic protocol within the SCSI device that can be used by client computer systems to provide cluster-wide locking semantics. All of the complexities of the locking are implemented in the initiators. Therefore, future changes or enhancements to the locking semantics are implemented by modifying client software rather than modifying and retrofitting SCSI storage device firmware.

This specification is written from the perspective of implementing cluster locking services, though SCSI Memory Export isn't limited to just that. Any application that requires memory to be shared between initiators on a SCSI bus or SAN can benefit from the Memory Export commands.

2 Memory Export Concepts

2.1 Lock Buffers

The SCSI storage device with SCSI Memory Export provides buffers of memory to a of cluster machines. Each buffer is used as a commonly-accessible memory area to store state associated with particular lock.

2.2 Buffer IDs

The Buffer ID (BID) is the 9-byte unsigned integer tag used to identify a buffer. Since the buffer space is sparse, the BID will be mapped to a physical buffer using a hash table or other lookup scheme. The actual value of the BID is opaque to the DMEP device. It is just an arbitrary bit-string to be used as a label for buffers.

2.3 Physical Buffer Number

The physical buffer number is an index associated with each buffer on the storage device. This number never changes, even if the Buffer ID changes. This number can be the order in which the buffers are allocated, or their order in memory, but it is not required. All that is required is that the total buffers on the storage device have physical buffer numbers from 0 to N-1, where N is the number of buffers, and that the physical buffer numbers never change while the storage device is on.

Physical Buffers may be added to a storage device, while running. These added buffers have Physical Buffer Numbers numbered starting at the last buffer number plus one (N) to the new number of physical buffers. Buffers may be added by vendor defined interfaces, and by Memory Export Selects.

2.4 Conditional Stores

The Memory Export specification defines a way to do an atomic read-modify-write operation on a buffer. This allows a cluster member to read in the lock state from a buffer, modify that lock state, and store it back out to the Memory

Export device. It is necessary, however, that the operation be implemented in two parts. There is a buffer load (read) and a buffer store (write). (The modify part happens within the initiator.)

It is possible, however, that between the time a computer reads a buffer, and when it stores the modified buffer, the buffer might have been changed by a second computer. This specification provides a mechanism which prevents the first computer from overwriting the second computer's buffer and hence corrupting the lock state in that buffer.

This mechanism is a conditional store. Each buffer contains an increasing 64-bit sequence number. If the sequence number passed to the drive as part of the store operation is not equal to the sequence number contained in the physical buffer, the store fails. The store operation also passes in the number of the physical buffer to be changed. If this number isn't the number of the physical buffer that the BID is mapped to, the store fails. This ensures that the Logical Buffer Number to Physical Buffer number association hasn't changed between the load and store operations.

Upon initial creation of each physical buffer, the sequence number should be set to an 64 bit pseudo random value.

2.5 In-Use and Just-Created Buffers

Each Memory Export Buffer device will support a number of buffers, some of which contain relevant data, while others contain unused data, or are unused. Buffer which have been successfully stored will take on the characteristic of being "In Use" until the are marked as being unused. If a Load command is issued on a buffer that is not In Use, then the buffer will take on the characteristic of being "Just Created". Just Created buffers remain on the device so that the Store command that follows the Load command can have a Sequence Number and Physical Buffer Number match. Two subsequent loads of Just Created Buffers that have not been made In Use by a successful store should have the same Buffer ID to Physical Buffer Number mapping so long as the buffer remains Just Created. Just Created Buffers can be removed from the device arbitrarily if the buffers are needed for more recently created buffers. In Use Buffers should not be reused until marked unused.

2.6 Buffer Segments

To offer greater flexibility and buffer isolation, some Memory Export devices may optionally implement multiple independent buffer spaces. These buffer segments operate in complete isolation from each other. The number of buffers and size of each buffer can be user specified for each buffer segment. Memory Export devices may implement a single buffer segment, or up to 256 buffer segments.

2.7 Full Buffer Space

Since the Memory Export device can support more Buffer IDs than it has actual buffer space, the buffers stored must be kept in a sparse buffer arrangement. Actual buffers in the memory of the storage device must be mapped to Buffer IDs dynamically. When all the available physical buffers are in use, the storage device is no longer able to map any more new buffers until some of the used ones are discarded. The storage device indicates, with every transaction, how full the actual buffer space is.

2.8 Enable

If a storage device should happen to be powered off or reset, and the state of the memory containing the buffers is lost, the initiators using Memory Export service must be informed. Before a segment will accept any buffer loads or stores, it must be enabled with a Memory Export Enable command. Therefore, when the storage device is reset, it will no longer accept buffer loads or stores. This protects any data locked by those buffers from accidental corruption.

3 The SCSI Memory Export Commands

The SCSI Memory Export Commands are generic, low latency operations which facilitate cluster locking among other services. There are two memory Export SCSI commands with five defined service actions to perform loads, stores, buffer dumps, configuration senses, and configuration selects. The two Memory Export SCSI commands are

Code	Service Action	Description
0	LOAD BUFFER	Read the state of a single Memory Export buffer off of the target
1	DUMP BUFFERS	Read a large chunk of the Memory Export buffer state off of the target
2	SENSE CONFIG	Read a segment Memory Export Configuration off of the target
3-31	Reserved	

Table 1: MEMORY EXPORT IN Service Actions

Byte,Bit	7	6	5	4	3	2	1	0
0	Operation Code (85h)							
1	Reserved			Service Action				
2	Segment Number							
3	(MSB)							
4								
5								
6				Buffer Number				
7								
8								
9								
10								
11								(LSB)
12								
13				Allocation Length				
14								
15	Control							

Table 2: Memory Export In CDB

MEMORY EXPORT OUT for service actions sending data to the SCSI device and MEMORY EXPORT IN for service actions pulling data off the SCSI device. The two SCSI command structures and five Services are shown below.

The procedure to change buffer data information is this: First the client issues the load command, fetching the buffer state into memory. The data returned includes a 64-bit sequence number and a 64-bit physical buffer number. The client then changes the buffer state (in a way that is outside the scope of this specification), and stores the changed buffer, passing with it the same physical buffer number and sequence number. If the sequence number on the buffer in the storage device is the same as the sequence number passed in by the client, and the physical buffer number for the buffer in the storage device is the same as the physical buffer number passed in by the client, then the store is successful. This means that the data portion of the buffer is copied into the appropriate location in the device and the sequence number for that buffer is incremented. If the store was not successful, a Check Condition status is returned.

4 The SCSI MEMORY EXPORT IN Command

The MEMORY EXPORT IN command causes the target to report data to the initiator. This data can be configuration information, the contents of a Memory Export buffer, or a large chunk of the Memory Export buffer space as it exists on the target. The MEMORY EXPORT IN command can return different data depending on the Service Action which is specified in the CDB. These Service Actions are shown in Table 1.

4.1 The Memory Export In CDB

The Memory Export In command has a 16 byte Command Descriptor Block (CDB). It is shown in Table 2.

4.1.1 Operation Code

The proposed SCSI Operation code for Memory Export In is 85h.

4.1.2 Service Action

This describes what Memory Export action is being used. Depending on the Service Action, some of the other fields of the CDB may have variable meanings.

If the Memory Export device receives a MEMORY EXPORT IN command with a Service Action larger than 2, a SCSI CHECK CONDITION of ILLEGAL REQUEST is returned with ASC/Q of INVALID FIELD IN CDB (24h/0h) and the sense key specific field set to CC0001h.

If the Memory Export device receives a MEMORY EXPORT IN command with a Service Action other than 2 addressing a segment which has not been configured, a SCSI CHECK CONDITION of ILLEGAL REQUEST is returned with ASC/Q of INVALID FIELD IN CDB (24h/0h) and the sense key specific field set to C00002h.

If the Memory Export device receives a MEMORY EXPORT IN command with a Service Action other than 2 addressing a segment which has not been enabled, a SCSI CHECK CONDITION of ILLEGAL REQUEST is returned with ASC/Q of MEMORY EXPORT SEGMENT NOT ENABLED (04h/0Ah) and the sense key specific field set to 0h.

4.1.3 Segment Number

The number of the segment in which the desired buffer resides.

4.1.4 Buffer Number

This is the 72-bit number of the buffer on which to operate. In the case of Memory Export Load service actions this represents the Logical Buffer ID. In the case of Memory Export Dump Buffer actions, this field targets a Physical Buffer number. Since Physical Buffer Numbers are 64 bit values, the first byte of this field is ignored on Memory Export Dump Buffer actions.

4.1.5 Allocation Length

The Allocation Length specifies the maximum number of bytes that the Memory Export device should return to the client.

4.2 The Memory Export Load Service Action and Parameter Data Format

The Load service action allows an initiator to read the current state of a buffer on the Memory Export device along with its associated header information. The header information is necessary if the initiator then tries to change the buffer state with a Store action. The Memory Export Load action uses the Memory Export In CDB as shown in table 2.

Service Action The Service Action code for the Load Buffer Command is 0.

Buffer Number The Buffer Number field is used to store the Buffer ID of the buffer which is to be returned to the initiator.

Segment Number The number of the segment in which the addressed buffers reside.

Allocation Length The maximum length, in bytes, to be returned. The Allocation Length shall be at least 3 bytes.

The Parameter Data Format for the Memory Export Load is shown in Table 3. The PDF of the load operation has exactly the same format as the PDF for the store operation. The contents of these fields are:

4.2.1 Length

The number of bytes that should be returned by a Load action. (24 + segment data size)

4.2.2 Service Action

The Service Action code for the Load Buffer Command is 0.

4.2.3 In Use

This field indicates whether a buffer is in use. Buffers not in use can be reclaimed by the storage device for use as other buffer IDs. In order to deallocate a buffer and make it eligible to be reclaimed by the storage device, a client system will issue a store command with the In Use bit set to zero.

4.2.4 Fullness

This field indicates what portion of the actual buffer space is filled. These buffers are marked with the 'In Use' Flag. The Fullness indicates how full the buffer space is scaled to fit in a 8-bit quantity, with FFh indicating the buffer space is completely full, and 00h indicating that none of the buffer space is used.

4.2.5 Physical Buffer Number

This field is a Physical Buffer Number for the buffer on the storage device in which the buffer data is stored.

4.2.6 Sequence Number

This number is incremented every time the buffer is successfully stored. This prevents a client computer from overwriting buffers which are in an unknown state. Upon initial allocation of a physical buffer, the sequence number should be set to a pseudo random value.

4.2.7 Data

This is the actual buffer which is being accessed. The storage device knows nothing about what is in the buffer. It is simply a bunch of bytes.

4.3 The Memory Export Dump Buffers Service Action

The Memory Export Dump Buffer Service Action is necessary for lock recovery. It simply dumps out buffers based on their Physical Buffer Number and not on the Buffer ID. The command returns a bunch of buffers starting with the Physical Buffer Number, and dumps as many buffers as it can fit within Allocation Length Bytes. The Dump Buffer command returns only In-Use buffers that reside in the segment number that is specified in the CDB. The MEMORY EXPORT IN CDB fields should be interpreted as follows, for all Dump Buffer commands.

Service Action The Service Action code for the Dump Buffer Command is 1.

Buffer Number The Buffer Number field is used to store the Physical Buffer Number at which the dump should start. The first byte of the Buffer number is ignored by the SCSI device in the case of Memory Export Dump Buffer actions.

If the Memory Export device receives a MEMORY EXPORT Dump Buffer command addressing a Physical Buffer Number which doesn't exist, a SCSI CHECK CONDITION of ILLEGAL REQUEST is returned with ASC/Q of INVALID FIELD IN CDB (24h/0h) and the sense key specific field set to C00004h.

Byte,Bit	7	6	5	4	3	2	1	0
0	(MSB)							
1				Length (n)				
2								(LSB)
3		Reserved			Service Action (0)			
4	In Use	Reserved						
5	Fullness							
6				Reserved				
7								
8	(MSB)							
...				Sequence Number				
15								(LSB)
16	(MSB)							
...				Physical Buffer Number				
23								(LSB)
24								
...				Data				
n								

Table 3: Memory Export Load Parameter Data Format

Segment Number The number of the segment in which the addressed buffers reside.

Allocation Length The maximum length, in bytes, that the device shall return. The Allocation Length should be at least 8 bytes.

The Parameter Data Format of the Dump Buffer Command is show in table 4. Its parts are:

4.3.1 Returned Byte count

The number of bytes that are returned by the Dump PDF. This will usually be the Allocation Length rounded down to match the size of returned buffers. However, if the returned Physical Buffers are at the end of the storage device's list of buffers, or not many buffers are in use on the storage device, there may not be enough remaining buffers to fill the requested allocation length. Therefore, this number could be far smaller than the allocation length specified in the CDB.

It is very important that the writers of initiator software always set the Allocation Length to at least 8 to get at least these first 8 header-like bytes. If the allocation length is not large enough to return a single buffer, zero buffers are returned.

The first physical buffer that is returned in the PDF will be the first "In Use" buffer having a physical buffer number that is greater than or equal to the physical buffer number that is specified in the CDB.

If the more bit in the PDF is set, this indicates that the entire CDB allocation length has been filled with returned buffers, and that there are more buffers on the storage device that can be dumped by issuing a subsequent dump command. Another dump command should be issued specifying the physical buffer number of the last buffer that was returned in the PDF plus one.

If the more bit is not set, then there are no more buffers to be returned with a buffer number greater than the last buffer number that was returned in the PDF. In this case, another dump command is not necessary.

Also, if the returned byte count is 8, there are no more buffers to be returned with a buffer number that this is greater than or equal to the buffer number specified in the CDB. In this case, the more bit will also be zero.

4.3.2 service Action

The Service Action code for the Dump Buffer Command is 1.

4.3.3 More

This bit indicates that there are additional In Use physical buffers on the Segment beyond the last buffer returned in the Dump PDF.

4.3.4 Buffer ID 1

The Buffer ID of the first Physical Buffer returned.

4.3.5 Physical Buffer Number 1

The Physical Buffer Number of the first Physical buffer returned.

4.3.6 Sequence Number 1

The Sequence Number of the first Physical buffer returned.

4.3.7 Data 1

The Buffer Data of the first Physical Buffer returned.

4.3.8 Following Fields

The first buffer returned is the buffer with the Physical Buffer Number addressed by the CDB. The following buffers are the numerically sequential Physical Buffers, based on the Physical Buffer Number. Only In-Use buffers are returned in the event of a Dump Buffers command. At the beginning of every buffer header is a reserved field. This ensures that the dump buffer PDF is 32-bit aligned in all cases that the buffer size is even.

4.4 The Sense Configuration Service Action

The Memory Export Protocol is a very generic and versatile Protocol. Various users of the protocol might have very different needs for the size of each buffer, and the number of buffers that they use. For that reason, each segment can be configured separately. The Memory Export Sense Configuration action reports the current configuration of a segment to the initiator.

The format of the Memory Export Sense Configuration command is shown below in table 5. The Memory Export In CDB for a Sense Configuration command is interpreted as follows.

Service Action The Service Action code for the Sense Configuration Command is 2.

Buffer Number The Buffer Number field is ignored for Sense Configuration Commands.

Segment Number The number of the segment for which the configuration information should be returned.

Allocation Length The maximum length, in bytes, that the device shall return.

4.4.1 Length

This is the length of the SENSE CONFIGURATION PDF.

Byte,Bit	7	6	5	4	3	2	1	0
0	Returned Byte Count							
2								
3	Reserved			Service Action (1)				
4	More	Reserved						
5	Reserved							
7								
8	Reserved							
10								
11	Buffer ID 1							
19								
20	Sequence Number 1							
27								
28	Physical Buffer Number 1							
35								
36	Data 1							
...								
n								
n+1	Reserved							
n+3								
n+4	Buffer ID 2							
n+12								
...	...							
...								
...	Data M							
...								
m*n								

Table 4: Dump Buffer PDF

Byte,Bit	7	6	5	4	3	2	1	0
0	(MSB)	Length						
1								
2								(LSB)
3	Reserved			Action (2)				
4	Number of Segments							
5	Maximum Supported Segments							
6	Reserved							
7								
8	(MSB)	Number of Buffers for Segment						
...								
15								(LSB)
16	(MSB)	Data Size for Segment						
17								
18								(LSB)
19	Reserved							

Table 5: Memory Export Sense Configuration Parameter Data Format

Code	Service Action	Description
0	STORE BUFFER	Change the state of a single Memory Export buffer on the target
1	Reserved	
2	SELECT CONFIG	Set the Memory Export Configuration of the target
3	ENABLE SEGMENT	Activate a Memory Export segment on the target
4-31	Reserved	

Table 6: MEMORY EXPORT OUT Service Actions

4.4.2 Service Action

The Service Action code for Sense Configuration is 2.

4.4.3 Number of Segments

The number of segments on the Memory Export device currently configured with at least 1 buffer of size greater than 0 bytes. These are not necessarily sequential, and do not necessarily start with segment 0.

4.4.4 Maximum Supported Segments

The Maximum number of Segments supported by the Memory Export device minus one. (For example, a Maximum Supported Segments field of 0Fh indicates that 16 segments are supported.)

4.4.5 Number of Buffers

The number of Memory Export buffers in the segment identified by the Segment Number in the CDB.

4.4.6 Data Size

The length, in bytes, of each Memory Export Buffer in the segment identified by the Segment Number in the CDB.

5 The MEMORY EXPORT OUT Command

The MEMORY EXPORT OUT command allows the intiator to configure a Memory Export device, and to write buffers to the device. The allowed service actions for the MEORY EXPORT OUT command are show in table 6.

5.1 The Memory Export Out CDB

The parts of the Memory Export Out CDB are shown in Table 7, they are:

5.1.1 Operation Code

The proposed SCSI Operation code for Memory Export Out is 89h.

5.1.2 Service Action

This describes what Memory Export action is being used. The possible values for this are shown in Table **??**.

5.1.3 Segment Number

The number of the segment in which the desired buffer resides.

Byte,Bit	7	6	5	4	3	2	1	0
0	Operation Code (89h)							
1	Reserved			Service Action				
2	Segment Number							
3	(MSB)							
4								
5								
6				Buffer Number				
7								
8								
9								
10								
11								(LSB)
12								
13				Parameter Length				
14								
15	Control							

Table 7: Memory Export Out CDB

5.1.4 Buffer Number

This 72-bit number of the buffer to be acted upon. In the case of Stores this indicates the logical buffer number of the buffer on which to operate.

5.1.5 Parameter Length

The Parameter Length specifies the number of bytes that the initiator sends to the Memory Export Device.

If the Memory Export device receives a MEMORY EXPORT OUT command with a Service Action of 1 or larger than 3, a SCSI CHECK CONDITION of ILLEGAL REQUEST is returned with ASC/Q of INVALID FIELD IN CDB (24h/0h) and the sense key specific field set to CC0001h.

If the Memory Export device receives a MEMORY EXPORT OUT command with a Service Action other than 2 addressing a segment which has not been configured, a SCSI CHECK CONDITION of ILLEGAL REQUEST is returned with ASC/Q of INVALID FIELD IN CDB (24h/0h) and the sense key specific field set to C00002h.

5.2 The Memory Export Store Service Action and Parameter Data Format

The Store action allows an initiator to write state to a new buffer, or to modify the state of an existing buffer. For a Store action to succeed, the initiator must provide input parameters that match the output header information returned by a Load action. Otherwise a SCSI check condition is returned. The Store action uses the Memory Export Out CDB.

Service Action The Service Action code for the Store Buffer Command is 0.

Buffer Number The Buffer Number field is used to store the Buffer ID of the buffer which is addressed by the initiator.

If the Memory Export device receives a MEMORY EXPORT Store command addressing a Buffer that is not In Use or Just Created, a SCSI CHECK CONDITION of ILLEGAL REQUEST is returned with ASC/Q of BUFFER ID NEVER LOADED (26h/10h) and the sense key specific field set to C00003h.

Segment Number The number of the segment in which the addressed buffers reside.

If the Memory Export device receives a MEMORY EXPORT Store command addressing a segment which has not been enabled, a SCSI CHECK CONDITION of ILLEGAL REQUEST is returned with ASC/Q of MEMORY EXPORT SEGMENT NOT ENABLED (04h/0Ah) and the sense key specific field set to 0h.

Parameter Length The length, in bytes, of the Parameter data sent to the Memory Export device. This field should be 24 bytes plus the segment data size when a buffer is being stored, and 24 bytes when a buffer is being freed.

If the In Use Bit is set and the Parameter Length is not equal to the data size for the addressed segment plus 24, a SCSI CHECK CONDITION of ILLEGAL REQUEST is returned with ASC/Q of PARAMETER LIST LENGTH ERROR (1Ah/0h) and the sense key specific field set to 800000h.

If the In Use Bit is not set and the Parameter Length is not equal to 24, a SCSI CHECK CONDITION of ILLEGAL REQUEST is returned with ASC/Q of PARAMETER LIST LENGTH ERROR (1Ah/0h) and the sense key specific field set to 800000h.

The Parameter Data format for the Memory Export Store action is shown in Table 8. The PDF for the Store operation is the same format as the PDF for the Load operation, except that the Fullness field is ignored. The fields of these formats are:

5.2.1 Length

The number of bytes sent in the PDF.

5.2.2 Service Action

The Service Action code for the Store Buffer Command is 0.

5.2.3 In Use

This field indicates whether a buffer is in use. Buffers not in use can be reclaimed by the storage device for use as other buffer IDs. In order to deallocate a buffer and make it eligible to be reclaimed by the storage device, a client system will issue a store command with the In Use bit set to zero.

5.2.4 Physical Buffer Number

This field is a Physical Buffer Number for the buffer on the storage device in which the buffer data is stored. If this does not match the PBN associated with the PBN on the Memory Export Device, a SCSI CHECK CONDITION of MISCOMPARE is returned with ASC/Q of BUFFER NUMBER ERROR (26h/0Fh).

5.2.5 Sequence Number

This number is incremented every time the buffer is successfully stored. This prevents a client computer from overwriting buffers which are in an unknown state. Upon initial allocation of a physical buffer, the sequence number should be set to a pseudo random value. If this does not match the Sequence Number associated with the Buffer Number on the Memory Export Device, a SCSI CHECK CONDITION of MISCOMPARE is returned with ASC/Q of SEQUENCE NUMBER ERROR (26h/0Eh).

5.2.6 Data

This is the actual buffer which is being accessed. The storage device knows nothing about what is in the buffer. It is simply a bunch of bytes.

Byte,Bit	7	6	5	4	3	2	1	0
0	(MSB)							
1				Length (n)				
2								(LSB)
3		Reserved			Service Action (0)			
4	In Use	Reserved						
5				Reserved				
7								
8	(MSB)							
...				Sequence Number				
15								(LSB)
16	(MSB)							
...				Physical Buffer Number				
23								(LSB)
24								
...				Data				
n								

Table 8: Memory Export Store Parameter Format

5.3 The Memory Export Select Configuration Command

The Memory Export Protocol is a very generic and versatile Protocol. Various users of the protocol might have very different needs for the size of each buffer, and the number of buffers that they use. For that reason, each segment can be configured separately. The Memory Export Select action changes the configuration of a segment.

Segments are unconfigured at power-on time for the Memory Export Device. They effectively have 0 buffers of size 0 bytes. If an unconfigured segment is targeted by a Select Config command, it is configured and can be subsequently enabled and used. If an already configured segment is targeted by a Select Config command with the same Buffer Size and Number of Buffers dimensions as it previously supported, the contents of the buffers are cleared, and all are marked as unused. If an existing segment is the target of a Select Config command with differing dimensions, then the existing buffers are all removed, and new buffers of the new size are allocated.

In the event that a segment is targeted with a Select Configuration command which attempts to allocate more buffers than can be allocated from the available resources on the Memory Export device, the device creates as many buffers as possible of the Buffer Size requested in the Parameter Data. If there are not enough resources available for even a single buffer of the requested Buffer Size, then 0 buffers are created. If the Select command indicates 0 buffers of size 0 bytes, then the segment is returned to the unconfigured state.

Whenever a segment's configuration has changed, it cannot be addressed by Memory Export Load, Store, or Dump commands until it has been enabled with the Memory Export Enable action. This prevents buffer size from changing and state from being removed, without the initiators being aware of the change.

When a Memory Export Select Configuration command is issued to the Memory Export device, the segment addressed in the MEMORY EXPORT OUT CDB will be configured per the parameter data. The format of the Memory Export Select command parameter data is shown below in table 9. The MEMORY EXPORT OUT CDB fields should be interpreted as follows for all Select Configuration Commands.

Service Action The Service Action code for the Select Configuration Command is 2.

Buffer Number The Buffer Number field is ignored for Select Configuration Command.

Segment Number The number of the segment for which the configuration information should be changed.

Parameter Length The length, in bytes, of the parameter data. If the Parameter Length is not 18 bytes, a SCSI

Byte,Bit	7	6	5	4	3	2	1	0
0	(MSB)							
1				Length				
2								(LSB)
3		Reserved			Service Action (2)			
4				Reserved				
5								
6				Reserved				
7								
8	(MSB)							
...				Number of Buffers for Segment				
15								(LSB)
16	(MSB)							
17				Data Size for Segment				
18								(LSB)
19	Reserved							

Table 9: Memory Export Select Parameter Data Format

CHECK CONDITION is returned with ASC/Q of PARAMETER LIST LENGTH ERROR (1Ah/0h) and the sense key specific field set to 800000h.

5.3.1 Length

This field is the number of bytes sent in the PDF.

5.3.2 Service Action

The Service Action code for Select Configuration is 2.

5.3.3 Number of Buffers

The number of Memory Export buffers in the segment identified by the Segment Number in the CDB. If the Number of Buffers feild is 0 and the Data size field is not 0, a SCSI CHECK CONDITION is returned with ASC/Q of INVALID FIELD IN PARAMETER LIST (26h/0h) with the sense key specific field set to 800008h.

5.3.4 Data Size

The length, in bytes, of each Memory Export Buffer in the segment identified by the Segment Number in the CDB. If the Data Size feild is 0 and the Number of Buffers field is not 0, a SCSI CHECK CONDITION is returned with ASC/Q of INVALID FIELD IN PARAMETER LIST (26h/0h) with the sense key specific field set to 800010h.

5.4 The Enable Segment command

The Enable Service Action activates a segment so that it may receive Memory Export LOAD, STORE, and DUMP actions. Any attempt to Issue these commands to segments that have not been enabled will return a SCSI check condition. A segment may not be enabled unless it has been configured by a SELECT CONFIGURATION first. Any Enable command received on a non-configured segment will return a SCSI check condition. The MEMORY EXPORT OUT CDB feilds should be interpreted as follows for all Enable Segment commands.

Condition	Condition handling
Power Cycle	Issue Unit Attention: *Power On* for all initiators All buffers are zeroed
SCSI Reset *	Buffers not affected
Bus Device Reset *	Buffers not affected
Task Management ** Target Reset	Buffers not affected

Table 10: Memory Export Power Cycles, and Resets: * Parallel SCSI only, ** Fibre Channel only

Sense Key	Additional Sense	ASCQ	Description
05h	04h	0Ah	Memory Export segment not enabled
05h	26h	10h	BID never loaded
0Eh	26h	0Eh	Sequence Number Error
0Eh	26h	0Fh	Buffer Number Error
06h	2Ah	06h	Memory Export Parameters Have Changed

Table 11: SCSI Check Conditions Sense Qualifiers for Memory Export Out commands

Service Action The Service Action code for the Enable Segment Command is 3.

Buffer Number The Buffer Number field is ignored for the Enable Segment Command.

Segment Number The number of the segment which is to be Enabled.

Parameter Length The length, in bytes, of the parameter data, which is 0.

5.5 Exception Handling

SCSI resets shall not clear Memory Export Buffers. This is shown in Table 10.

6 RAID Controller Sparse Memory Export Space Implementation

Modern RAID Controllers provide the ability to support up to one gigabyte or more of installed memory. This large memory space is typically used by the RAID controller as a logical block data cache. RAID controllers can theoretically preallocate a portion of this memory space to provide a large number of Memory Export buffers. Within most RAID controller implementations, it should be possible to provide hundreds of thousands of physical buffers with virtually no impact on RAID controller performance. The following discussion describes one approach to sparse Memory Export Buffer implementation within a RAID controller. Other implementations are possible that may be more suited to different RAID controller architectures.

6.1 Processing Memory Export commands

When a storage device receives a Memory Export Load command, it must first look through it's table of "In Use" Memory Export Buffers to see if the Buffer ID is already in use. If the BID is already in use the buffer is returned to the initiator.

If the BID is not already in use, then a buffer must be removed from the free list. Since a store operation has not yet been performed, and the "In Use" flag is not set, the buffer is still not really used. However, the BID must be concretely mapped to a Physical Buffer ID so that the subsequent Memory Export Store command can succeed.

Therefore, the Buffer is placed on the “Just Created” list. This list can be a linear list, a hash table, or any other quick look-up data structure.

When a Buffer in the Just Created List is acted on by a Memory Export Store, and the In-Use bit is set, the buffer is moved to the regular “in use” buffer table. If a buffer in the Just Created list is not addressed for a period of time it must be returned to the free buffers list. This prevents the just created list from growing too large and increasing the duration of searching for Buffers. Also, if there are no available buffers on the free buffers list, the storage device may reclaim buffers from the “Just Created” list.

The “Just Created” list can be avoided if the entire free list is implemented as a hash table, thus reducing the time to look up a Buffer ID. Each chain, or perhaps the whole table, must be reused in Least Recently Used order, so that machines can send a store after a load, before the hash table is emptied. In this case it is not necessary to implement a timeout strategy in order to remove stale buffers from a just created list.

Alternately a just created list can be a small hash table with few buckets. When a load command is received and the free list is empty, a buffer may be pulled out of the just created list and associated with the new Buffer ID supplied in the load command. Note that the buffer will lose it’s original Buffer ID association. For this reason, the least recently used buffer should be targeted for this reassignment. The buffer is then placed back into the appropriate hash bucket of the just created list.

6.2 Mapping of Buffer IDs to Physical Buffers

Buffer IDs (BIDs) that are passed in the CDB to the storage device must be mapped to physical buffer structures within the storage device. If the Buffer structure does not exist, it must be created by the storage device. The BID is stored to enable lookup of the buffer structure. Subsequent Memory Export operations can then look up the physical buffer structure by using the BID that is input in the CDB.

6.3 Buffer Space Initial Allocation

The number of buffers and their size is set by the Memory Export Select Config as is shown in Table 9. If the product of these two variables exceeds the memory allowed by the RAID controller, the Data Size must obey the mode select if possible, and the number of buffers will be as large as is possible given the constraint of Data Size. The RAID device should have an interface with which the user can set how much of the device’s memory should be available for Memory Export buffers. This interface is up to the implementor and should be similar to other configuration methods available on that storage device. The device can optionally set a limit to how much memory can be made available for Memory Export buffers.

6.4 Buffer Structure Organization

Upon completion of power-up, or after receipt of a Select Config command to set Memory Export parameters, the RAID controller creates a free list that contains memory that is used for dynamic creation of buffer structures. This buffer structure consists of the BID, sequence number, buffer data, and a link pointer to link the structure into a hash table. The hash table is simply an array of pointers that point to a linked list of buffer structures. Buffer structures are inserted into the linked list at the appropriate array index that is derived as a result of mapping the BID to a hash table index. The implementation of a physical Buffer space is shown in figure 1.

6.5 Efficiency of Memory Allocation

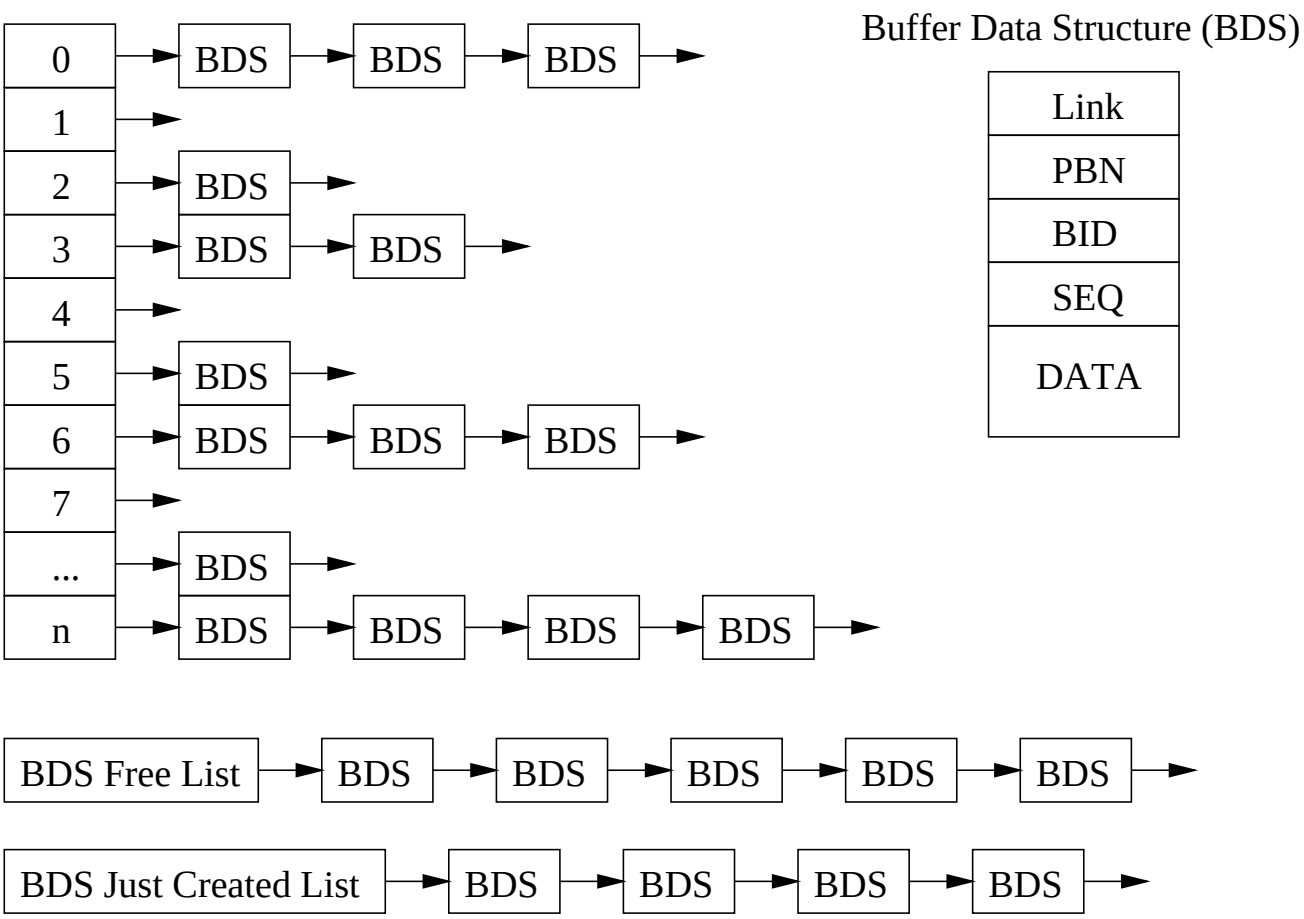
In order to provide good performance, Memory Export operations need to be executed with minimal latency. Therefore, allocation and deallocation of buffer structures should be performed very efficiently. Functions that perform heap allocation such as malloc and free should be avoided due to extremely high overhead. Preallocation of memory to support the required number of buffers, or a fast slab memory allocation scheme should be considered.

6.6 Buffer Data Structure Deallocation

A physical buffer structure can only be deallocated if it is marked as unused. Upon deallocation, the buffer structure is removed from the in use buffer table, or just created list, and placed on the free list. The operation of moving the buffer structure from the in use buffer table to the free list consists of a remque to remove the buffer from the in use buffer table, followed by an insque to insert the buffer on the free list. This is a very low overhead operation and therefore may be preferable to leaving the buffer data structure in the table and reusing or reclaiming it at a later time.

6.7 Full Memory Export Space

In the event that a Memory Export operation is requested and it is not possible to allocate a buffer structure, the RAID controller will normally indicate this by setting the “full” field in the PDF to indicate that the memory export buffer space is full and that memory resources are not available to satisfy the request. The Memory Export action will fail. Returning a “full” field of 0xFF, and zero for all other PDF fields. Note that the “Full Memory Export Space” condition is reported on a per segment basis. Upon noticing the “full” indication in the PDF, the client will issue an out of band requests to other clients requesting that they drop buffers. The clients will then release buffers by marking them unused, and the buffer space of the storage device will return to a “not full” condition. The original client will then re-issue the request that previously failed.



> Each BDS contains a link, Physical Buffer Number (PBN), Buffer ID (BID), Sequence Number (SEQ), and Data.
> To locate a BDS, the storage device will hash the BID supplied by the CDB to find the hash table index, and then traverse the list until the BDS is found.
> If the BDS is not found, the Just Created List is searched.
> If the BDS is still not found, one is allocated from the free list and placed on the Just Created List.
> A newly created BDS is moved from the Just Created List to the hash table upon receipt of a successful store command with the In Use bit set.
> To improve performance the Just Created List can also be implemented as a hash table.
> The BDS will be removed from the hash table or the Just Created List on a successful store on that BID with the In Use bit set to 0.

Figure 1: A depiction of physical buffer data structures stored in a hash table.

7 Acknowledgments

Many people have contributed ideas to The DLOCK Specification and Memory Export Specification over the years. The people we can remember at the moment are:

- From **Sistina Software**
  Jonathan E. Brassow, Erling Nygaard, David C. Teigland, Michael C. Tilstra
- From the **University of Minnesota**
  Russell Cattelan, Grant M. Erickson, Benjamin I. Gribstad, Thomas M. Ruwart, Aaron Sawdey,
- From **Compaq**
  Richard Lary
- From **Seagate Technology, Inc.**
  Dave Anderson, Jim Coomes, Tony Hecker, Gerry Houlder, Nate Larson, Michael H. Miller, Troy Wheeler
- From **NASA Ames Research Center**
  Alan Poston, John Lekashman
- From **Ciprico, Inc.**
  Raymond Gilson, Edward A. Soltis
- From **Novell, Inc.**
  Robert Wipfel
- From **Oracle Corp.**
  David Brower

References

[1] Kenneth Preslan et al. *Proposed SCSI Device Locks Version 0.9.4*. University of Minnesota, Parallel Computer Systems Laboratory, http:// www.globalfilesystem.org/ pubs/ dlock-0.9.4.ps, 1999.

[2] Andrew Barry and Kenneth Preslan et al. *Proposed SCSI Device Locks Version 0.9.5*. University of Minnesota, Parallel Computer Systems Laboratory, http:// www.globalfilesystem.org/ pubs/ dlock-0.9.5.ps, 1999.

[3] Andrew Barry et al. *Proposed SCSI Device Locks Version 0.9.5B*. University of Minnesota, Parallel Computer Systems Laboratory, http:// www.globalfilesystem.org/ pubs/ dlock-0.9.5B.ps, 1999.

[4] Andrew Barry et al. *Proposed SCSI Device Locks Version 0.9.6*. University of Minnesota, Parallel Computer Systems Laboratory, http:// www.globalfilesystem.org/ pubs/ dlock-0.9.6.ps, 2000.

[5] Kenneth Preslan et al. A 64-bit, shared disk file system for linux. In *The Sixteenth IEEE Mass Storage Systems Symposium held jointly with the Seventh NASA Goddard Conference on Mass Storage Systems & Technologies*, San Diego, California, March 1999.

[6] Kenneth Preslan et al. Implementing journaling in a linux shared disk file system. In *The Seventeenth IEEE Mass Storage Systems Symposium held jointly with the Eigth NASA Goddard Conference on Mass Storage Systems & Technologies*, New York, New York, March 2000.

[7] X3T10 SCSI committee. Document T10/ 98-225R1 – Proposed SCSI Device Locks. http:// ftp.symbios.com/ ftp/ pub/ standards/ io/ x3t10/ document.98/ 98-225r1.pdf, October 1998.

[8] Roy G. Davis. *VAXCluster Principles*. Digital Press, 1993.

[9] Kenneth Preslan, Steven Soltis, Christopher Sabol, and Mat thew O'Keefe. Device locks: Mutual exclusion for storage area networks. In *The Seventh NASA Goddard Conference on Mass Storage Systems and Technologies in cooperation with the Sixteenth IEEE Symposium on Mass Storage Systems*, San Diego, CA, March 1999.

[10] Andrew Barry, Kenneth Preslan, and Matthew O'Keefe. An overview of version 0.9.5 proposed scsi device locks. In *The Eighth NASA Goddard Conference on Mass Storage Systems and Technologies in cooperation with the Seventeenth IEEE Symposium on Mass Storage Systems*, New York, NY, March 2000.