TO: T10 Membership
FROM: Paul A. Suhler, Seagate Removable Storage Solutions, LLC
DATE: 5 March 2003
SUBJECT: T10/03-077r3, ADI Bridging Proposal

Revision 3:

- Revised ADC model to change concept of partitioned SMC device server into two independent device servers with different sets of mandatory commands.
- Banned splitting the Bridging Login IU across multiple frames.
- Clarified that the bridging manager may initiate commands to remote device server while operating in passthrough bridging mode, e.g., to refresh cached data.
- To support caching of inquiry, etc. data by the data transfer device, an asynchronous notification IU from the automation to the drive is now defined.

Revision 2:

- In Bridging Login IU, changed LU Index back to LUN.
- Added padding field to SCSI Request IU.

Revision 1:

- The feature is renamed to "bridging" with "passthrough" and "hosted" variants.
- ADC text no longer refers to ADT.
- Added a model section for ADT. This refers to ADC.
- Changed "frame" references to "information unit."
- 4.2.x: Sentence with "typically" was rewritten as an example.
- "Surrogate mode" field in mode page was changed to an enabled bit plus one bit for each type of bridging. To determine what types are supported, automation can request the changeable mode page. (The alternative is trial and error setting of bits.) Only one type bit shall be set.
- Put back in text saying that if bridging is disabled, then that LUN is not reported.
- "Process login" is changed to "Bridging Login." This makes the intent of the IU apparent to a reader of the ADT standard.
- Moved the ACCEPT bit to byte 0, bit 7 in bridging login and logout IUs.
- Added a LOGICAL UNIT INDEX field to the login IU just in case we need to support bridging for multiple LUs in the automation device. Other text still refers specifically to the SMC device server. Do we want to make it more general at this time? Future changes to support multiple bridged LUs of types other than SMC will not require IU changes, just rewording of the standard.

Revision 0:

Specification of ADI Bridging Operation requires changes to:

- ADC model to describe bridging operation
- ADC automation drive descriptor to select different bridging modes
- ADT link services clause to add bridging login and logout
- ADT SCSI Request frame payload to add an I_T nexus identifier

FCP-2 doesn't have an entry in the definitions clause for process login, so I haven't provided one here for bridging login.

Thanks to Paul Entzel for providing large parts of this text. I wordsmithed some of his text, so blame me for any problems.

The following is to be added to the ADC model section:

4.2.x ADI bridging operation

The data transfer device may optionally support one or more forms of ADI bridging operation for the automation device. When this operation is enabled, the data transfer device reports a logical unit to its primary interface ports that implements an SMC device server, and the automation device reports a logical unit to its ADT port that implements an SMC device server. In the process of executing a SCSI command or task management request, the data transfer device's SMC device server (called the "local device server") may invoke execution of a command or request by the automation device's SMC device server (called the "remote device server"). The entity within the local device server that performs invocation of commands or requests on the remote device server is called the bridging manager.

The effect is that some or all commands and requests addressed to the local device server are passed to the remote device server through the ADT port. This can be used, for example, in low-cost automation devices that do not have separate primary interface ports.

A data transfer device that supports the ADI bridging feature may either pass all commands to the remote device server ("passthrough bridging"), or it may support a subset of the SMC and SPC command sets ("hosted bridging") and pass the remainder to the remote device server. For instance, in the latter case a REPORT LUNS command may be processed by the local device server instead of being passed to the remote device server. This may be necessary if the local device server has direct knowledge of the other LUNs supported by the target device. The Medium Changer descriptor in the ADC Target Device Configuration mode page allows for multiple types of ADI bridging operation to be enabled in the data transfer device. Support for any particular type of ADI bridging operation is optional.

4.2.x.1 Hosted bridging operation

In hosted bridging mode, the remote device does not necessarily implement the full set of SPC and SMC commands. No indication of the original initiator port of the SCSI Request is passed between the data transfer and automation devices. In this mode of operation, the local device server is capable of servicing commands and task management functions that require knowledge of the originating initiator port. Effectively, the data transfer device acts as a protocol bridge in this mode.

The local device server shall support commands as required by SPC-3 and SMC-2.

If any of the following commands are supported, they shall be serviced by the local device server and not passed through to the remote device server:

a) RESERVE(6) and RESERVE(10)
b) RELEASE(6) and RELEASE(10)
c) PERSISTENT RESERVE IN
d) PERSISTENT RESERVE OUT
e) REPORT LUNS
f) REQUEST SENSE

The local device server shall also perform the following actions:

a) Check for reservation conflicts on all commands routed to the remote device server. Return RESERVATION CONFLICT on all commands that violate a reservation condition and not pass them through to the SMC device.

b) Manage UNIT ATTENTION conditions generated for multiple initiators. Before passing a command through to the remote device server, pending UNIT ATTENTION conditions shall be checked. If the local device server detects a UNIT ATTENTION condition is pending for an initiator port when a new command is received from it, the local device server shall return Check Condition for the command without passing it to the remote device server.
c) When the primary interface uses contingent allegiance, sense data must be saved on a per initiator port basis.

Remote device servers which support hosted bridging shall support the commands specified in Table x:

Table x – Commands for remote SMC device servers in hosted bridging

Command	Required	Reference
CHANGE DEFINITION	O	SPC-3
EXCHANGE MEDIUM	O	SMC-2
INITIALIZE ELEMENT STATUS	O	SMC-2
INITIALIZE ELEMENT STATUS WITH RANGE	O	SMC-2
MOVE MEDIUM	M	SMC-2
POSITION TO ELEMENT	O	SMC-2
REQUEST VOLUME ELEMENT ADDRESS	O	SMC-2
REPORT SUPPORTED OPERATION CODES	O	SPC-3
SEND VOLUME TAG	O	SMC-2

4.2.x.2 Passthrough bridging operation

In passthrough bridging operation, for each command or request received by the local device server, the bridging manager shall invoke that command on the remote device server. The commands are passed from the data transfer device to the SMC device server with an identifier for the I_T nexus from which the command was received. The details of how I_T nexus identification is performed and how commands, data, and status are passed between the devices is described in the appropriate transport protocol.

Passthrough bridging operation does not preclude the bridging manager's initiating commands for the remote device server. These commands do not require identification of an I_T nexus. For example, this may be done to obtain inquiry data for caching, as described below.

Remote device servers which support passthrough bridging shall support commands as required by SPC-3 and SMC-2.

4.2.x.3 Caching SMC data

In some implementations, the local device server may cache some data from the remote device server. For instance, the local device server may save the inquiry data from the remote device server and return it to any initiator port that requests it. The ADT standard provides means for the remote device server to notify the local device server when data that may be cached has changed. The local device server should discontinue using cache data from the remote device server once it is informed of a possible change until such time as it can be refreshed.

The following change is for ADC clause 6.2.2.3.3, Medium Changer descriptor parameters:

- The one-bit ENABLE field is supplemented with HOST BRIDGING MODE and PASSTHROUGH BRIDGING MODE fields and supporting text is added.

Table 29 – Medium changer descriptor

Byte \ Bit	7	6	5	4	3	2	1	0
6	Reserved					HBM	PBM	ENABLED
7	Reserved							

If the HOSTED BRIDGING MODE (HBM) field is set to one, then the data transfer device shall not identify the I_T nexus of each bridged command to the automation device.

If the PASSTHROUGH BRIDGING MODE (PBM) field is set to one, then the data transfer device shall identify the I_T nexus of each bridged command to the automation device.

The HBM and PBM fields shall not both be set to one at the same time. This avoids ambiguity about which bridging mode is to be used.

If the ENABLED field is set to one it indicates the Logical Unit is reported and supported. Commands received for this logical unit shall be passed on to the automation using the ADT interface. When it is set to zero, the logical unit is not reported to a REPORT LUNS command and does not respond to commands.

If the ENABLED field is changed from one to zero or if both the HBM and PBM fields are set to zero, then the data transfer device shall implicitly abort all commands issued to the automation device and shall report a status of CHECK CONDITION with a sense key of COMMAND ABORTED and an additional sense code of I_T NEXUS LOSS OCCURRED for each command. An implicit bridging logout shall be performed. All remaining device servers in the data transfer device shall report a change in the logical unit inventory, as specified in SPC-3, to any application clients connected through a primary interface port.

The following is to be added to the ADT model section in clause 4:

The ADT link and transport layers implement support for ADI bridging operations. The link layer contains link services to perform the bridging login and logout functions, which register and deregister I_T nexuses with the automation device when the passthrough bridging mode is active. In the protocol layer, the SCSI Request information unit (see 7.1.1) contains an I_T nexus identifier which identifies the I_T nexus which was the source of the command.

ADT bridging functions are controlled by the Medium Changer descriptor in the ADC device configuration mode page. In that descriptor, the ENABLED field enables and disables bridging, while the HOSTED BRIDGING MODE (HBM) and PASSTHROUGH BRIDGING MODE (PBM) fields select which type of bridging is used. Either the HBM or PBM field may be set to one, but not both.

When the HOSTED BRIDGING MODE (HBM) field is set to one, the data transfer device shall not perform a bridging login with the automation device. Each SCSI Request information unit shall contain a value of zero in its I_T NEXUS IDENTIFIER field.

When the PASSTHROUGH BRIDGING MODE (PBM) field is set to one, the data transfer device shall perform a bridging login with the automation device for each initiator from which it receives commands for the medium changer logical unit. Each SCSI Request information unit shall contain the I_T nexus identifier value established in that initiator's bridging login. Commands initiated by the data transfer device's bridging manager do not require a bridging login.

If the ENABLED field is changed from one to zero or if both the HBM and PBM fields are set to zero, then the data transfer device shall perform an implicit bridging logout.

The following changes are for ADT clause 6.5, Link service information units:

1. Add two new values to Table 4.
2. Add max payload size requirement to Port Login IU for support of bridging.
3. Add descriptive text and tables for bridging login and logout.

Table 4 – Link service information units

Payload Type	Description
0h	ACK (acknowledge)
1h	NAK (negative acknowledge)
2h	Port login
3h	Port logout
4h	Pause
5h	NOP (no operation)
6h	Initiate recovery
7h	Bridging login
8h	Bridging logout
9h – Fh	Reserved

The following paragraph in clause 6.5.3, Port Login IU, is modified. Note correction to font size of first letters in field name.

The MAXIMUM PAYLOAD SIZE value indicates the maximum number of bytes in the payload of a frame that the port can accommodate. A port must be capable of supporting a frame payload size of at least 270 bytes to support this transport layer. A port must be capable of supporting a frame payload size of at least 515 bytes to support bridging login.

6.5.9 Bridging login

Bridging Login information units are sent by the data transfer device to the automation device to identify an initiator port to the SMC device server. Bridging Login IUs are sent by the automation device to the data transfer device to accept or reject the bridging login. Automation devices shall transmit Bridging Login IUs only in response to receiving a Bridging Login IU.

Before the data transfer device performs bridging login, the data transfer device shall perform port login with the automation device and the PBM and ENABLED fields shall be set to one in the Medium Changer descriptor of the ADC device configuration mode page.

If the automation device transmits a Port Logout information unit, then all bridging logins are implicitly terminated.

Table x defines the payload of the Bridging Login information unit:

Table x – Bridging Login IU payload contents

Byte \ Bit	7	6	5	4	3	2	1	0
0	ACCEPT	Reserved						
1	I_T NEXUS IDENTIFIER							
2	RELATIVE PORT IDENTIFIER							
3	INITIATOR PORT IDENTIFIER LENGTH (m)							
4	INITIATOR PORT NAME LENGTH (n)							
5	(MSB)	INITIATOR PORT IDENTIFIER						
4+m								(LSB)
5+m	(MSB)	INITIATOR PORT NAME						
4+m+n								(LSB)

The ACCEPT bit shall be set to zero in the bridging login payload transmitted by the data transfer device. If the automation device supports passthrough bridging, then it shall set the ACCEPT bit in the Bridging Login IU payload it transmits.

The I_T NEXUS IDENTIFIER field contains a value identifying the SCSI Port of an application client sending a request to the SMC device server. This value shall not be zero.

The RELATIVE PORT IDENTIFIER field indicates which primary interface port the initiator is connected through. The relative port identifier value shall be one of the values returned in the Device Identifier VPD page, as specified in SPC.

Editor's Note: The relative port identifier field in that VPD page is four bytes. Is it a problem to reduce it to one byte? Is the last sentence even necessary?

The INITIATOR PORT IDENTIFIER field contains the port identifier of the SCSI Port of the application client. The INITIATOR PORT IDENTIFIER LENGTH field contains the length in bytes of the INITIATOR PORT IDENTIFIER field.

The INITIATOR PORT NAME field contains the port name of the SCSI Port of the application client. The INITIATOR PORT NAME LENGTH field contains the length in bytes of the INITIATOR PORT NAME field.

6.5.10 Bridging logout

Bridging Logout information units are sent by the data transfer device to the automation to terminate a bridging login. Upon receiving a Bridging Logout IU (see Table x), the automation device shall abort all open exchanges associated with that login. It shall then transmit a Bridging Logout IU to the data transfer device and shall transmit no further SCSI encapsulation IUs for any exchanges associated with that login.

Automation devices shall transmit Bridging Logout IUs only in response to receiving a Bridging Logout IU.

Table x - Bridging Logout IU payload contents

Byte \ Bit	7	6	5	4	3	2	1	0
0	ACCEPT	Reserved						
1	I_T NEXUS IDENTIFIER							

The ACCEPT bit shall be set to zero in the bridging logout payload transmitted by the data transfer device. If the automation device has an existing login identified by the value in the I_T NEXUS

IDENTIFIER field, then it shall set the ACCEPT bit to one in the bridging login IU payload it transmits. If it has no such login, then it shall set the ACCEPT bit to zero.

The I_T NEXUS IDENTIFIER field contains the I_T nexus identifier for the initiator port being logged out.

The following changes are for ADT clause 7.1.2 SCSI Request information unit:

- The I_T NEXUS and COMMAND REFERENCE NUMBER fields are added to the SCSI Request information unit payload in Table 8.
- The paragraphs describing the I_T NEXUS and CRN fields are added following the paragraph describing the BUFFER ALLOCATION LENGTH field.

Table 8 – SCSI Request IU payload contents

Byte \ Bit	7	6	5	4	3	2	1	0
0	LUN							
1								
2	TASK MANAGEMENT FUNCTION							
3	Reserved							
4	CDB							
19								
20	(MSB) BUFFER ALLOCATION LENGTH							
23								(LSB)
24	I_T NEXUS IDENTIFIER							
25	COMMAND REFERENCE NUMBER							

If the SCSI Request IU is transmitted by the automation device, then the I_T NEXUS IDENTIFIER field shall contain zero. If the SCSI Request information unit is being transmitted by the data transfer device as part of SCSI Passthrough bridging operation, then the I_T NEXUS IDENTIFIER field shall contain the value in the I_T NEXUS IDENTIFIER field in the Bridging Login request IU for the initiator originating the SCSI request. If the request was originated by the bridging manager, then the I_T NEXUS IDENTIFIER field shall contain zero.

If the SCSI Request information unit is transmitted by the data transfer device as part of ADI hosted bridging operation, then the I_T NEXUS IDENTIFIER field shall contain zero.

The COMMAND REFERENCE NUMBER (CRN) field contains the number sent by the initiator of a bridged command to assist in performing precise delivery checking for commands. If precise delivery checking is enabled, a nonzero value in the CRN field shall be treated as a command reference number in determining the receipt and ordering of commands from a particular initiator to the particular logical unit. If precise delivery checking is enabled, a zero value in the CRN field indicates that the command shall not be verified for precise delivery. If precise delivery checking is not enabled, the CRN field shall be ignored by the device server. If the SCSI Request information unit specifies a task management function, the CRN field shall be reserved and set to zero and the SCSI Request information unit shall not be verified for precise delivery.

If the SCSI Request IU is transmitted by the automation device, then the CRN field shall be zero.

Editor's Note: I left the Reserved field in byte 3 so that there would be room for future expansion. Should we instead move either I_T NEXUS IDENTIFIER or CRN to that byte and add future bytes at the end?

The following changes are made to clause 7.2.4, AER information unit, to support notification of possible changes to cached data:

Asynchronous Event Report IUs may optionally be used to report that an event has occurred that may be of interest. ~~Only a Data Transfer device may initiate AER IUs.~~ The payload of an AER IU transmitted by a data transfer device shall contain the VHF Polling data as defined in ADC.

The payload of an AER IU transmitted by an automation device shall contain the payload specified in Table x below. It shall be transmitted only when bridging operation is enabled and when a change has occurred to data in a mode page, log page, or vital product data page. The payload consists of a list of the numbers of changed pages for each type of page.

Table x – Automation AER IU payload

Byte \ Bit	7	6	5	4	3	2	1	0
0	CHANGED MODE PAGE LIST LENGTH (m)							
1	CHANGED LOG PAGE LIST LENGTH (n)							
2	CHANGED VPD PAGE LIST LENGTH (o)							
3	number of first changed mode page							
	…							
2+m	number of last changed mode page							
3+m	number of first changed log page							
	…							
2+m+n	number of last changed log page							
3+m+n	number of first changed VPD page							
	…							
2+m+n+o	number of last changed VPD page							

The data transfer device's bridging manager is not required to request any or all of the changed pages reported. It may request pages not reported as changed and may request standard inquiry data.

Editor's Notes:

1. Is this level of detail overkill? We could instead go with no payload and let the DTD ask for what it needs. Or we could just have one bit for mode, log, and VPD with no further detail.

2. We could get very detailed here and invent some control mechanism to let the DTD tell the automation device which data should generate an AER. I don't think that's necessary.
