TO: T10 Membership
FROM: Paul A. Suhler, Certance
DATE: 6 May 2003
SUBJECT: T10/03-077r7, ADI Bridging Proposal

Revision 7:

- Various changes from meeting.

Revision 6:

- Added definitions.
- Expanded discussion of caching of data in local device server in order to dovetail with description of NOTIFY DTD command.
- Incorporated changes in wording of clause 7.1.4, ADT rev. 4.

Revision 5:

- Deleted Automation AER IU in favor of NOTIFY DATA TRANSFER DEVICE (separate proposal 03-165r0).
- Prohibit support by the local device server of element reservations or of ELEMENT_SCOPE.
- Added discussion of enabling/disabling to the model section at the end of the local device server description.

Revision 4:

- Merged both forms of bridging into one.
- Reorganized AER IU. AER IU from automation is named "Automation AER IU" and the one from DTD is "DTD AER IU."
- Added four-byte First Burst Length field to SCSI Request IU for data-in commands.
- Not included:
  - Restricting non-LUN 0 device servers from supporting transport-related mode and log pages.
  - Renaming automation AER IU to DTD AER IU.

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 definitions clause:

3.1.a Accessible state: The state of a device server in which it would respond to a command with any combination of status and sense key other than CHECK CONDITION and NOT READY. If the device server would respond to a command with a status of CHECK CONDITION and sense key of NOT READY, then it is in the non-accessible state.

3.1.b Automation application client: In an automation device, the entity that performs invocation of commands or requests on the ADC device server in the data transfer device.

3.1.c Bridging manager: In a data transfer device implementing bridging, the entity that performs invocation of commands or requests on the remote SMC device server.

3.1.d Local SMC device server: The SMC device server in a data transfer device implementing bridging.

3.1.e Remote SMC device server: The SMC device server in an automation device which receives SCSI commands and task management requests via a data transfer device implementing bridging.

3.1.f Removable medium commands (RMC) device server: A generic term for the device server in a removable medium data transfer device installed in an automation system, e.g., an SSC (SCSI stream commands) device.

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

New paragraph following 4.2.1, last bullet item:

These operations are performed by invoking various SCSI commands and task management requests on the ADC device server. The application client within the automation device that invokes these requests is called the "automation application client" (see 3.1.b) Communications between the SMC device server and the automation application client are performed by means outside the scope of this standard.

4.2.2.x ADI bridging

4.2.2.x.1 ADI bridging introduction

The data transfer device may optionally support ADI bridging for the automation device. When this operation is enabled, the data transfer and automation devices shall contain the objects shown in Figure x, including those marked as optional. The data transfer device reports to its primary interface ports a logical unit that implements an SMC device server (called the "local SMC device server"), and the automation device reports a logical unit to its ADT port that implements an SMC device server (called the "remote SMC device server").

The local SMC device server will receive a SCSI command or task management request via a DTD primary port. In processing the command or request, it may require the automation device to perform one or more operations or to provide information. To do this, it passes requests to an application client in the data transfer device, called the "bridging manager." This communication is performed by means outside the scope of this standard. Using the ADT ports on the DTD and automation device, the bridging manager then invokes requests on the remote SMC device server which resides in the automation device.

The effect is that some or all requests addressed to the local SMC device server are passed to the remote SMC 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.

4.2.2.x.2 Local SMC device server operation

The local SMC device server shall support commands as required by the SCSI Medium Changer device type. Because the remote SMC device server lacks information about the initiator port which originated a request, it cannot implement the full set of commands. Thus, the local SMC device server shall service commands and task management functions that require knowledge of the originating initiator port. Effectively, the data transfer device acts as a protocol bridge.

If any of the following commands are supported, they shall be executed by the local SMC device server and not passed through to the remote SMC 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 SMC device server shall not support element reservations in the RESERVE(6), RELEASE(6), RESERVE(10), and RELEASE(10) commands. It shall not support the ELEMENT_SCOPE in the PERSISTENT RESERVE IN and PERSISTENT RESERVE OUT commands.

The local SMC device server shall also perform the following actions:

a) Check for reservation conflicts on all commands. Return RESERVATION CONFLICT on all commands that violate a reservation condition.
b) Manage UNIT ATTENTION conditions generated for multiple initiators. If the local SMC device server detects that a UNIT ATTENTION condition is pending for an initiator port when a new command is received from it, the local SMC device server shall return Check Condition for the command.
c) When the primary interface uses contingent allegiance, save sense data on a per initiator port basis.

The remote SMC device server shall not report any protocol-specific mode pages.

4.2.2.x.3 Bridging manager operation

ADI bridging is enabled and disabled via the Medium Changer Logical Unit mode descriptor implemented by the ADC device server (see clause 6.2.2.3.3). The descriptor specifies the logical unit number of the local SMC device server. When bridging is disabled, the logical unit shall not be reported to a REPORT LUNS command and the local SMC device server shall not respond to commands.

If the bridging manager receives any device service responses with a status of CHECK CONDITION and sense key of UNIT ATTENTION, it shall discard the response and reissue the command. All other responses with a status of CHECK CONDITION, including those with a sense key of NOT READY, shall be returned to the local SMC device server for subsequent return via the DTD primary port. This shall have no effect on the cached ready status, as described in 4.2.2.x.4.

The bridging manager shall operate in a single threaded fashion, i.e., not issue more than one request at a time to the remote SMC device server. For this reason, queued requests received via the DTD primary port shall be queued in the local SMC device server and issued to the bridging manager one at a time. Moreover, if execution of a single request by the local SMC device server requires issuing multiple requests to the remote SMC device server, those requests shall be issued one at a time to the bridging manager.

4.2.2.x.4 Caching SMC data and status

In some implementations the local SMC device server may preserve some data or status from the remote SMC device server in a cache, in order to respond quickly to certain commands without need for the bridging manager to invoke a command on the remote SMC device server. For instance, it may save the inquiry data from the remote SMC device server and return it to any initiator port that requests it.

Caching of SMC status, inquiry data, and mode data is controlled by the CACHE field in the Medium Changer descriptor (see 6.2.2.3.3). When the CACHE field is set to one, caching is enabled and the automation application client shall invoke the NOTIFY DATA TRANSFER DEVICE command (5.x) on the ADC device server when events occur that may change data cached by the local SMC device server. When the local SMC device server becomes aware of a possible change in the cached data, it shall discontinue using the cached data until it has been refreshed. It shall issue the necessary refresh commands to the bridging manager before issuing any commands that it may have received from a DTD primary port and queued.

If caching is disabled, then the automation application client need not invoke NOTIFY DATA TRANSFER DEVICE for purposes of indicating changes in cached data. It may still invoke the command to notify the DTD of events not related to changes in cached data.

Ready status indicates whether the remote SMC device server is accessible. When the remote SMC device server would respond to a command with a status of CHECK CONDITION and report a sense key of NOT READY, the remote SMC device server is not accessible. Otherwise, it is accessible. The local SMC device server may monitor the ready status of the remote SMC device server via the cache. If the ready status indicates not accessible, the local SMC device server shall report a status of CHECK CONDITION to commands requiring that the remote SMC device server be accessible, including TEST UNIT READY. It shall set the Sense Key to NOT READY and the additional sense code to that contained in the cache.

The automation application client shall invoke the NOTIFY DATA TRANSFER DEVICE command when any of the events that it reports have occurred since the previous invocation of a NOTIFY DATA TRANSFER DEVICE command. Moreover, the command shall report only those events that have not been previously reported.

Editor's Note: Should the above paragraph be moved to the description of NOTIFY DATA TRANSFER DEVICE?

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

- Add a one-bit field "CACHE" in bit 1, byte 6 of Table 29 – Medium changer descriptor.
- The descriptive text following Table 29 – Medium changer descriptor is changed as follows:

If the ENABLE field is set to one, it indicates that the Logical Unit is reported and supported. Commands received for this logical unit shall either be processed by the local SMC device server or passed by the bridging manager to the remote SMC device server. When it is set to zero, the logical unit is not reported in response to a REPORT LUNS command and it does not respond to commands.

If the ENABLE field is changed from one to zero, then the local SMC device server shall implicitly abort all commands in its task set and shall report a status of CHECK CONDITION with a sense key of COMMAND ABORTED and an additional sense code of LOGICAL UNIT COMMUNICATION FAILURE for each command. All remaining device servers in the data transfer device shall report a change in the logical unit inventory, as specified in SPC-2, to any application clients connected through a primary interface port.

If the device server receives a MODE SELECT command via a primary interface port, and the parameter data would change the ENABLE field, then it shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN PARAMETER LIST.

If the CACHE field is set to one, the local SMC device server shall implement caching of SMC data and status (see 4.2.2.x.4). Enabling of caching requires enabling of bridging; if the ADC device server receives a MODE SELECT command for which the parameter data would set the ENABLE field to zero and the CACHE field to one, then it shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN PARAMETER LIST.

If the CACHE field is set to zero, the local SMC device server shall not implement caching of SMC data and status.

The following changes are for ADT:

- 7.1.2, Table 8 – addition of first burst length field for data-in commands, to facilitate throttling data-in transfers:

Byte \ Bit	7	6	5	4	3	2	1	0
0	(MSB)	LUN						
1								(LSB)
2	TASK MANAGEMENT FUNCTION							
3	Reserved							
4	CDB							
19								
20	(MSB)	BUFFER ALLOCATION LENGTH						
23								(LSB
24	(MSB)	FIRST BURST LENGTH						
27								(LSB)

When the SCSI Request IU specifies a data-in command, the FIRST BURST LENGTH field shall contain a value indicating the amount of buffer space prepared in the initiator for the first SCSI Data IU and requesting the transfer from the target of one or more IUs of that length. This has the effect of a Transfer Ready IU with a BUFFER OFFSET field of zero and a BURST LENGTH field of FIRST BURST LENGTH. When the SCSI Request IU specifies a non-data or data-out command, the FIRST BURST LENGTH field shall contain zero.

- 7.1.3 SCSI Response IU – add new response code value to Table 11:

Value	Description
06h	Command complete with GOOD status and sense data valid
07h – FFh	Reserved

The response code value of "Command complete with GOOD status and sense data valid" shall be sent by the remote SMC device server when bridging is enabled and a command completes with a SCSI status of GOOD that will generate a unit attention to initiator ports other than the one that initiated the command. In this case, the SCSI STATUS field shall contain zero and the SCSI AUTOSENSE DATA field shall contain the sense data to be reported to those other initiator ports. Additionally, the data cached by the local SMC device server shall be invalidated (see ADC).

- 7.1.4 SCSI Transfer Ready IU description – generalized for both data-in and data-out:

A SCSI Transfer Ready IU shall be sent by ~~a target of an exchange~~ one port to inform the ~~exchange initiator~~ other port that it is ready to receive data associated with the command. The ~~target port~~ sender of the SCSI Transfer Ready IU may request all of the data associated with a command with a single SCSI Transfer Ready, or it may use multiple SCSI Transfer Ready IUs within the exchange context to request the data a little bit at a time. The contents of the SCSI Transfer Ready IU payload are described in Table 19.

The BUFFER OFFSET field indicates the offset from the beginning of the buffer associated with the first byte that shall be sent. Data shall not be requested out of order. This field can be used to recover from an error detected in transmission by allowing the ~~SCSI target port~~ receiver of the data to request re-transmission of the previous burst of data.

The BURST LENGTH field indicates the size of the buffer that has been allocated to receive data within the ~~target device~~ sender of the SCSI Transfer Ready IU. The ~~exchange initiator~~ receiver of the SCSI Transfer Ready IU shall respond ~~to the SCSI Transfer Ready IU~~ by transmitting data using one or more SCSI Data IUs until Burst Length bytes of data have been sent.