TO: T10 Membership
FROM: Paul A. Suhler, Certance
DATE: 2 May 2003
SUBJECT: T10/03-077r6, ADI Bridging Proposal

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

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

3.1.d Remote 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.

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

4.2.x ADI bridging

4.2.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 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.

4.2.x.2 Local device server operation

The local device server shall support commands as required by the SCSI Medium Changer device type. Because the remote device server lacks information about the initiator port which originated a command, it cannot implement the full set of commands. Thus, the local 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 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 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 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 and not pass them through to the SMC device.
b) Manage UNIT ATTENTION conditions generated for multiple initiators. If the local device server detects that 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 invoking a command on the remote device server.
c) When the primary interface uses contingent allegiance, save sense data on a per initiator port basis.

The local device server shall not report any protocol-specific mode pages. This avoids race conditions and conflicts in which a port's characteristics are changed simultaneously by MODE SELECT commands to different logical units.

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 device server. When bridging is disabled, the logical unit is not reported to a REPORT LUNS command and it does not respond to commands.

4.2.x.3 Caching SMC data and status

In some implementations the bridging manager may cache some data from the remote device server, in order to enable the local device server to respond quickly to certain commands without need for the bridging manager to invoke a command on the remote device server. For instance, it may save the inquiry data from the remote device server and return it to any initiator port that requests it. The automation device may invoke the NOTIFY DATA TRANSFER DEVICE command (5.x) on the ADC device server in order to notify the bridging manager that data that

may be cached has changed. When the bridging manager is informed of a possible change in the cached data, it shall discontinue using the cached data until it has been refreshed.

The bridging manager may also cache the ready status of the remote device server. The ready status indicates whether the remote device server is accessible. When the device would respond to a command with a status of CHECK CONDITION and report a sense key of NOT READY, the device is not accessible. Otherwise, it is accessible.

The automation device may 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 automation device shall report only those events that have not been previously reported.

Caching is controlled by the CACHE field in the Medium Changer descriptor (see 6.2.2.3.3). If caching is disabled, then the automation device need not invoke NOTIFY DATA TRANSFER DEVICE for purposes of informing the DTD of changes in cached data. (The command may still be invoked to notify the DTD of events not related to changes in cached data.)

When the remote device server executes a command, the automation device may invoke one or more NOTIFY DATA TRANSFER DEVICE commands on the ADC device server. If it were to invoke a new NOTIFY DATA TRANSFER DEVICE command before receiving status on the previous one, then a race condition could ensue if the bridging manager were to refresh cached data by invoking INQUIRY, MODE SENSE, or other commands on the remote device server. To avoid such a race condition or resource exhaustion via mutually recursive command invocation, the automation device shall not invoke any commands on the ADC device server until it has received status for the NOTIFY DATA TRANSFER DEVICE command.

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 device server or passed on by the bridging manager to the remote device server. When it is set to zero, the logical unit is not reported in response to a REPORT LUNS command and does not respond to commands.

If the ENABLE field is changed from one to zero, then the local 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, then the data transfer device shall implement caching of SMC data and status (see 4.2.x.3). Enabling of caching requires enabling of bridging; if the 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.

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:

Bit / Byte	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 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.

- 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.