ENDL
TEXAS

Date: 24 February 2003
To: T10 Technical Committee
From: Ralph O. Weber
Subject: Calls and Arguments in SAM-3

A clear and consistent nomenclature is proposed for the procedure calls contained in the SAM-3 architectural model.

Today, the terms 'call', 'procedure call', and 'remote procedure call' are used interchangeably. Likewise, the terms 'argument' and 'parameter' are used interchangeably.

The following two terms are proposed to be the only terms used in SAM-3:

- 'procedure call' — replacement for 'call' and 'remote procedure call'
- 'procedure parameter — replacement for 'argument' and 'parameter'
  (The use of 'procedure parameter' is intended to differentiate the parameters associated with procedure calls from the parameters found in parameter data.)

An additional proposal that is an option for the working group to consider is replacing the use of bold text for procedure call and procedure parameter names with dark colored text. In this proposal, any text cited for modification demonstrates this change using Dark Brown for procedure call names and Dark Green for procedure parameter names. Only the cited text reflects this change. Changing all procedure call and procedure parameter names is left as an exercise for the editor (in the event that the working group wishes to proceed with making the change).

Specific changes are proposed for SAM-3 r05.

It must be noted that all command set and SCSI transport protocol standards make reference to the procedure call and procedure parameter (i.e., argument) concepts in SAM. Most include a Notation For Procedure Calls subclause such as the SAM-3 subclause modified by Change 3. If these changes are approved for SAM-3, equivalent changes will be necessary in every SCSI standard that references SAM-3.

Revision History

r0 Initial proposal

18484 Preston Road, Suite 102 PMB 178, Dallas, TX 75252 214-912-1373 972-596-2775 Fx ENDL_TX@computer.org

Specific Changes:

Change 1 [glossary]: Convert the current 'call' glossary entry to 'procedure call' and add a glossary entry for 'procedure parameter' as follows.

Modify the 'call' glossary entry as follows (relocating it for proper alphabetical sorting):

> ~~**3.1.9**~~ **3.1.x procedure call:** An architectural abstraction having the appearance of a programming language function call that is used to model service interfaces. ~~The act of invoking a procedure.~~

Add a 'procedure parameter' glossary entry as follows:

> **3.1.y procedure parameter:** A datum provided as input to or output from a procedure call (see 3.1.x) architectural abstraction.

Change 2 [Editorial Conventions]: Modify the second and third paragraphs of 3.4 (Editorial Conventions) as follows:

> Upper case is used when referring to the name of a numeric value defined in this specification or a formal attribute possessed by an entity. When necessary for clarity, names of objects, ~~procedures~~ procedure calls, procedure parameters or discrete states are capitalized ~~or set in bold type~~. Names of fields are identified using small capital letters (e.g., NACA bit).
>
> The names of procedure calls ~~Callable procedures~~ are identified by ~~a name in bold~~ dark brown type, such as Execute Command (see clause 5). Names of ~~procedural arguments~~ procedure parameters are identified by dark green type. ~~denoted by capitalizing each word in the name.~~ For instance, Sense Data is the name of a procedure parameter ~~an argument~~ in the Execute Command procedure call.

Change 3 [Notation Conventions]: Modify 3.6.2 as follows:

3.6.2 Notation for procedure calls ~~procedures and functions~~

In this standard, the model for functional interfaces between entities is a procedure call (see 3.1.x). ~~the callable procedure.~~ Such interfaces are specified using the following notation:

[Result =] Procedure Name (IN ([Input-1] [,Input-2] …]), OUT ([Output-1] [,Output-2] …))

Where:

Result: A single value representing the outcome of the procedure call ~~or function~~.

Procedure Name: A descriptive name for the function modeled by the procedure call ~~to be performed~~. When the procedure call model is used to describe a SCSI transport protocol service, the procedure name is the same as the service name.

Input-1, Input-2, …: A comma-separated list of names identifying caller-supplied input ~~data~~ procedure parameters.

Output-1, Output-2, …: A comma-separated list of names identifying output ~~data~~ procedure to be returned by the procedure call.

"[…]": Brackets enclosing optional or conditional procedure parameters ~~and arguments~~.

This notation allows data procedure parameters to be specified as inputs and outputs. The following is an example of a procedure call specification:

Found = Search (IN (Pattern, Item List), OUT ([Item Found]))

Where:

Found = Flag

Flag, if set to one, indicates that a matching item was located.

Input Procedure Parameters ~~Arguments~~:

Pattern = … /* Definition of Pattern parameter */
Parameter containing the search pattern.

Item List = Item<NN> /* Definition of Item List as an array of NN Item parameters*/
Contains the items to be searched for a match.

Output Procedure Parameters ~~Arguments~~:

Item Found = Item … /* Item located by the search procedure */
This parameter is only returned if the search succeeds.

Change 4 [distributed service model]: In 4.2 (The SCSI distributed service model), make the following modifications:

> Client-server relationships are not symmetrical. A client may only originate requests for service. A server may only respond to such requests. The client calls the server-resident procedure and waits for completion. From the client's point of view, the behavior of a remote service invoked in this manner is indistinguishable from a ~~conventional~~ local procedure call. …

Change 5 [Execute Command]: In 5.1 (The Execute Command remote procedure), make the following modifications:

5.1 The Execute Command ~~remote~~ procedure call

An application client requests the processing of a SCSI command by invoking the SCSI transport protocol services described in 5.4, the collective operation of which is conceptually modeled in the following ~~remote~~ procedure call:

Service Response = Execute Command (IN (I_T_L_x Nexus, CDB, [Task Attribute], [Data-In Buffer Size], [Data-Out Buffer], [Data-Out Buffer Size], [Command Reference Number]), OUT ([Data-In Buffer], [Sense Data], [Sense Data Length], Status))

Input Procedure Parameters ~~Arguments~~:

I_T_L_x Nexus:	Either an I_T_L nexus or an I_T_L_Q nexus (see 4.12).
CDB:	Command descriptor block (see 5.2).
Task Attribute:	A value specifying one of the task attributes defined in 8.5. This procedure parameter ~~argument~~ shall not be specified for an untagged command or the second and subsequent commands in a sequence of linked commands. Untagged tasks shall implicitly have the SIMPLE attribute. The attribute of a task that processes linked commands shall be set according to the Task Attribute procedure parameter ~~argument~~ specified for the first command in the sequence.
Data-In Buffer Size:	The number of bytes available for data transfers to the ~~Data-In Buffer~~ data-in buffer (see 5.4.3).
Data-Out Buffer:	A buffer containing command specific information to be sent to the logical unit, such as data or parameter lists needed to service the command. The buffer size is indicated by the Data-Out Buffer Size procedure parameter ~~argument~~. The content of the ~~Data-Out Buffer~~ data-out buffer shall not change during the lifetime of the command (see 5.5) as viewed by the application client.
Data-Out Buffer Size:	The number of bytes available for data transfers from the ~~Data-Out Buffer~~ data-out buffer (see 5.4.3).
Command Reference Number (CRN):	When this procedure parameter ~~argument~~ is used, all sequential commands of an I_T_L nexus shall include a CRN procedure parameter ~~argument~~ that is incremented by one. The CRN shall be set to one for each I_T_L nexus involving the SCSI port after the SCSI port receives a hard reset or detects I_T nexus loss. The CRN shall be set to one after it reaches the maximum CRN value supported by the protocol. The CRN value zero shall be reserved for use as defined by the SCSI transport protocol. It is not an error for the application client to provide this procedure parameter ~~argument~~ when CRN is not supported by the SCSI transport protocol or logical unit.

Output Procedure Parameters ~~Arguments~~:

Data-In Buffer: A buffer to contain command specific information returned by the logical unit by the time of command completion. The Execute Command procedure call shall not return a status of GOOD, CONDITION MET, INTERMEDIATE, or INTERMEDIATE-CONDITION MET unless the buffer contents are valid. The application client shall not assume that the buffer contents are valid unless the command completes with a status of GOOD, CONDITION MET, INTERMEDIATE, or INTERMEDIATE-CONDITION MET. While some valid data may be present for other values of status, the application client should rely on additional information from the logical unit, such as sense data, to determine the state of the buffer contents. If the command ends with a service response of SERVICE DELIVERY OR TARGET FAILURE, the application client shall consider this procedure parameter to be undefined.

Sense Data: A buffer containing sense data returned concurrently with a CHECK CONDITION status (see 5.9.6). The buffer length is indicated by the Sense Data Length procedure parameter ~~argument~~. If the command ends with a service response of SERVICE DELIVERY OR TARGET FAILURE, the application client shall consider this procedure parameter to be undefined.

Sense Data Length: The length in bytes of the ~~Sense Data~~ sense data.

Status: A one-byte field containing command completion status (see 5.3). If the command ends with a service response of SERVICE DELIVERY OR TARGET FAILURE, the application client shall consider this procedure parameter to be undefined.

The Service Response procedure parameter assumes one of the following values:

TASK COMPLETE: A logical unit response indicating that the task has ended. The ~~status~~ Status procedure parameter shall have one of the values specified in 5.3 other than INTERMEDIATE or INTERMEDIATE-CONDITION MET.

LINKED COMMAND COMPLETE: Logical unit responses indicating that the task has not ended and that a linked command has completed successfully. As specified in 5.3, the ~~status~~ Status procedure shall have a value of INTERMEDIATE or INTERMEDIATE-CONDITION MET.

SERVICE DELIVERY OR TARGET FAILURE: The command has been ended due to a service delivery failure (see 3.1.113) or SCSI target device malfunction. All output procedure parameters are invalid.

Change 6 [Execute Command transport services]: Modify 5.4, 5.4.1, and 5.4.2 as follows:

5.4 SCSI transport protocol services in support of Execute Command

5.4.1 Overview

The SCSI transport protocol services that support the Execute Command ~~remote~~ procedure call are described in 5.4. Two groups of SCSI transport protocol services are described. The SCSI transport protocol services that support the request and confirmation for the Execute Command ~~remote~~ procedure call are described in 5.4.2. The SCSI transport protocol services that support the data transfers associated with processing a SCSI command are described in 5.4.3.

5.4.2 Execute Command request/confirmation SCSI transport protocol services

All SCSI transport protocol standards shall define the SCSI transport protocol specific requirements for implementing the Send SCSI Command SCSI transport protocol service request and the Command Complete Received confirmation. Support for the SCSI Command Received indication and Send Command Complete response by a SCSI transport protocol standard is optional. All SCSI I/O systems shall implement these SCSI transport protocols as defined in the applicable SCSI transport protocol specification.

SCSI Transport Protocol Service Request:

Send SCSI Command (IN (I_T_L_x Nexus, CDB, [Task Attribute], [Data-In Buffer Size], [Data-Out Buffer], [Data-Out Buffer Size], [Command Reference Number], [First Burst Enabled]))

Input Procedure Parameters ~~Arguments~~:

I_T_L_x Nexus:	Either an I_T_L nexus or an I_T_L_Q nexus (see 4.12).
CDB:	Command descriptor block (see 5.2).
Task Attribute:	A value specifying one of the task attributes defined in 8.5. For specific requirements on the Task Attribute procedure parameter ~~argument~~ see 5.1.
Data-In Buffer Size:	The number of bytes available for data transfers to the ~~Data-In Buffer~~ data-in buffer (see 5.4.3).
Data-Out Buffer:	A buffer containing command specific information to be sent to the logical unit, such as data or parameter lists needed to service the command (see 5.1). The content of the ~~Data-Out Buffer~~ data-out buffer shall not change during the lifetime of the command (see 5.5) as viewed by the application client.
Data-Out Buffer Size:	The number of bytes available for data transfers from the ~~Data-Out Buffer~~ data-out buffer (see 5.4.3).
Command Reference Number (CRN):	When this procedure parameter ~~argument~~ is used, all sequential commands of an I_T_L nexus shall include a CRN procedure parameter ~~argument~~ that is incremented by one (see 5.1).
First Burst Enabled:	~~An~~ A procedure parameter ~~argument~~ specifying that a SCSI transport protocol specific number of bytes from the ~~Data-Out Buffer~~ data-out buffer shall be delivered to the logical unit without waiting for the device server to invoke the Receive Data-Out SCSI transport protocol service.

SCSI Transport Protocol Service Indication:

SCSI Command Received(IN (I_T_L_x Nexus, CDB, [Task Attribute], [Command Reference Number], [First Burst Enabled]))

Input Procedure Parameters ~~Arguments~~:

I_T_L_x Nexus:	Either an I_T_L nexus or an I_T_L_Q nexus (see 4.12).
CDB:	Command descriptor block (see 5.2).
Task Attribute:	A value specifying one of the task attributes defined in 8.5. For specific requirements on the Task Attribute procedure parameter ~~argument~~ see 5.1.
Command Reference Number (CRN):	When this procedure parameter ~~argument~~ is used, all sequential commands of an I_T_L nexus shall include a CRN procedure parameter ~~argument~~ that is incremented by one (see 5.1).
First Burst Enabled:	~~An~~ A procedure parameter ~~argument~~ specifying that a SCSI transport protocol specific number of bytes from the ~~Data-Out Buffer~~ data-out buffer are being delivered to the logical unit without waiting for the device server to invoke the Receive Data-Out SCSI transport protocol service.

SCSI Transport Protocol Service Response (from device server):

Send Command Complete(IN (I_T_L_x Nexus, [Sense Data], [Sense Data Length], Status, Service Response))

Input Procedure Parameters ~~Arguments~~:

I_T_L_x Nexus:	Either an I_T_L nexus or an I_T_L_Q nexus (see 4.12).
Sense Data:	If present, this procedure parameter ~~argument~~ instructs the SCSI target port to return sense data to the SCSI initiator port (see 5.9.6).
Sense Data Length:	The length in bytes of the sense data to be returned to the SCSI initiator port.
Status:	Command completion status (see 5.1).
Service Response:	Possible service response information for the command (see 5.1).

SCSI Transport Protocol Service Confirmation:

Command Complete Received(IN (I_T_L_x Nexus, [Data-In Buffer], [Sense Data], [Sense Data Length], Status, Service Response))

Input Procedure Parameters ~~Arguments~~:

I_T_L_x Nexus:	Either an I_T_L nexus or an I_T_L_Q nexus (see 4.12).
Data-In Buffer:	A buffer containing command specific information returned by the logical unit on command completion (see 5.1).
Sense Data:	Sense data returned concurrently with a CHECK CONDITION status (see 5.9.6).
Sense Data Length:	The length in bytes of the received sense data.
Status:	Command completion status (see 5.1).
Service Response:	Service response for the command (see 5.1).

Note: The changes proposed above for 5.1 and 5.4 demonstrate the effect of this proposal on one subclause where 'procedure call' and 'procedure parameter' are in frequent usage. The remaining change paragraphs in this proposal list the other subclauses where similar changes are required. The effects of these changes will be like those shown above.

Change 7 ['remote procedure call']: Change 'remote procedure call' to 'procedure call' in the following subclauses:

- 4.2 (The SCSI distributed service model)
- 4.6.1 (The service delivery subsystem object)
- 5.4.3.1 (Data transfer SCSI transport protocol services introduction)
- 5.8.1 (Unlinked command example)
- 5.8.2 (Linked command example) [twice]
- 6.3.2 (Hard reset)
- 6.3.4 (I_T nexus loss)
- 7.10 (Task management SCSI transport protocol services)

Change 8 ['call']: Change 'call' to 'procedure call' in the following subclauses:

- 4.14 (The SCSI model for distributed communications)
- 5.4.2 (Execute Command request/confirmation SCSI transport protocol service) [unless 03-002 is approved before this proposal]

Change 9 ['argument']: Change 'argument' to 'procedure parameter' in the following subclauses:

- 4.3 (The SCSI client-server model)
- 4.15 (The SCSI model for distributed communications)
- 5.1 (The Execute Command remote procedure) [x2 (twice) in the last paragraph in subclause which is not shown in Change 5]
- 5.3.1 (Status codes) [CHECK CONDITION]
- 5.4.3 (Data transfer SCSI transport protocol services introduction) [3 times]
- 5.4.3.2 (Data-In delivery service) [twice]
- 5.4.3.3 (Data-Out delivery service) [4 times]
- 5.8.2 (Linked command example) [item 5]
- 6.4 (Event notification SCSI transport protocol services) [twice]
- 7 (Task management functions introduction)
- 7.10 (Task management SCSI transport protocol services) [5 times]

Change 10 ['parameter']: Change 'parameter' to 'procedure parameter' in the following subclauses:

- 3.1.12 (code value)
- 5.8.1 (Unlinked command example) [twice]
- 5.8.2 (Linked command example) [twice]
- 7.10 (Task management SCSI transport protocol services) [3 times]
- 7.11 (Task management function example) [item 3]

Change 11 ['Function call']: To both make the usage of 'procedure call' consistent and have a common nomenclature between clauses 5 and 7, change 'Function call:' to 'Request' in the following subclauses:

- 7.2 (ABORT TASK)
- 7.3 (ABORT TASK SET)
- 7.4 (CLEAR ACA)
- 7.5 (CLEAR TASK SET)
- 7.6 (LOGICAL UNIT RESET)
- 7.7 (QUERY TASK)
- 7.8 (TARGET RESET)
- 7.9 (WAKEUP)