

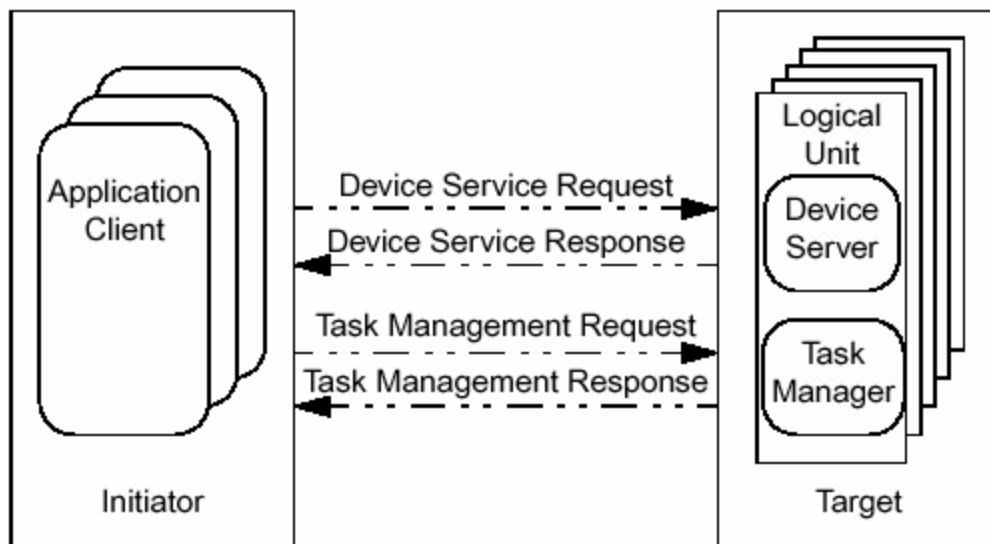
Overview

The SCSI architecture as specified in (see www.t101.org):

- *SCSI Architecture Model - 2 (SAM-2)*, Revision 18, dated 2001-May-31
- *SCSI Primary Commands - 3 (SPC-3)*, Revision 01, dated 2001-Sep-22
- *Fibre Channel Protocol for SCSI, Second Version (FCP-2)*, Revision 7, dated 2001-Mar-7
- *SCSI RDMA Protocol (SRP)*, Revision 10, dated 2001-Oct-3
- *IEEE Tutorial for SCSI use of IEEE company_id*, dated 1997-Feb-25
- *Access Controls for SPC-3*, dated 2001-Sep-22,
<ftp://ftp.t10.org/t10/document.01/01-268r2.pdf>.

SCSI Architecture (SAM-2)

In the Small Computer System Interface Architecture, a client originates requests and a server responds to them as shown in the next figure. SCSI commands are addressed to logical units, which contain a device server to process SCSI commands and a task manager to sequence and process task management requests. When a device containing at least one port originates SCSI commands and task management requests it is called an initiator. When a device containing at least one port contains logical units it is called a target. A SCSI domain contains at least one SCSI device, one initiator port, and one target port, interconnected by a service delivery subsystem through which application clients and device servers communicate. A target port contains a task router that sends tasks to logical units; if the LU is unknown the task is sent to LUN 0; if no LU is specified, the task management function is broadcast to all LUs known to the router. In I/O operation is a SCSI command, a series of linked SCSI commands, or a task management function.

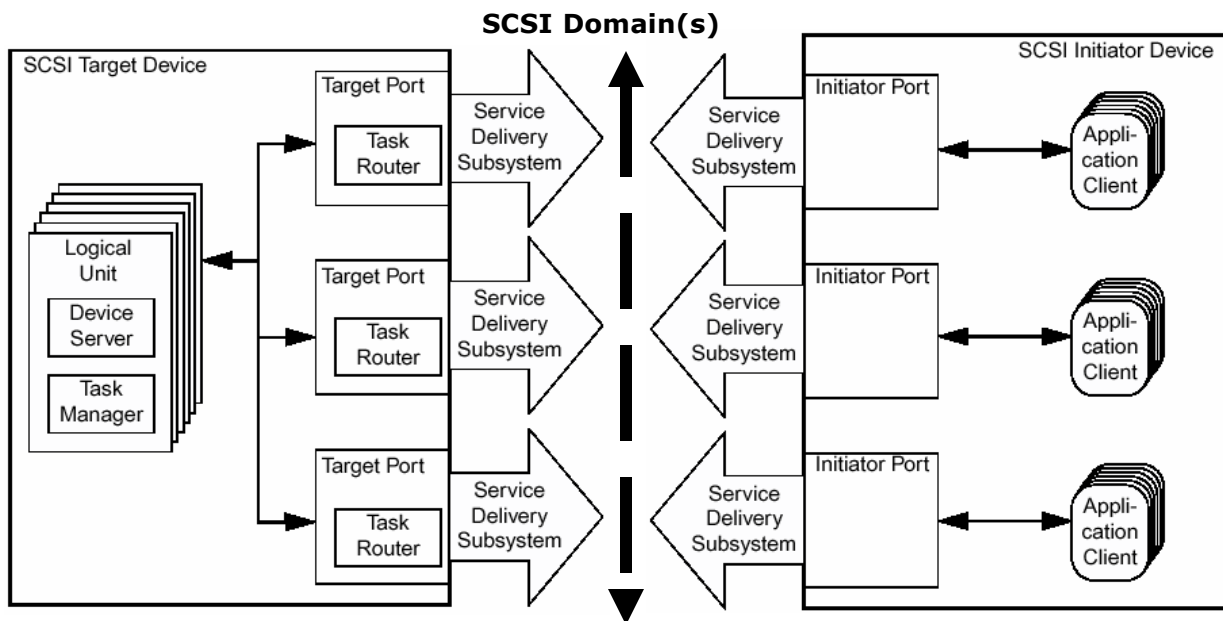


An application client, that is, a thread of processing within an initiator device such as a device driver, sends one or more SCSI commands to a target device. Requests, represented as tasks, are processed by the device server or task manager within the logical unit. A logical unit contains one or more task sets, each of which contain zero

or more tasks. A device server may reject commands that are not sent from the set of initiators determined by the application client using reservation commands.

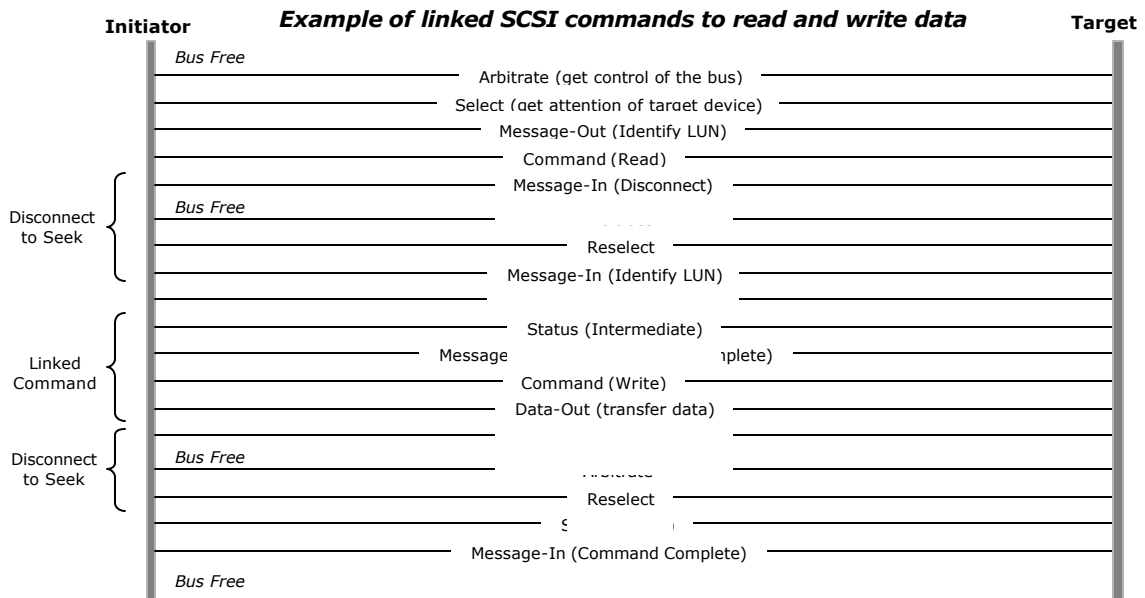
A logical unit may contain other logical units, referred to as dependent logical units. A device server that implements the hierarchical structure for dependent logical units sets the HISUP bit in the standard INQUIRY data returned by logical unit 0 (see SPC-2).

A SCSI device can have one or more ports as depicted in the following figure. Note that an initiator does not know if different target ports are in different devices. Likewise, a target does not know if different initiator ports are in different devices. However, application clients can discover that a logical unit is accessible via multiple target ports using the INQUIRY command vital product data page.



An application client sends requests by executing a remote procedure with eight input parameters: a nexus, a command descriptor block (CDB), a task attribute, number of bytes to transfer into the input buffer, an output buffer containing command-specific information, number of bytes to transfer from the output buffer, an option to automatically return sense data, and a command reference number. Output is returned in a buffer to hold command-specific information returned by the logical unit, a buffer to hold autosense data, and completion status.

A command executes in subsequent parts called phases, which may be interleaved with phases from other commands to other devices. Some phases are missing in some commands but the general sequence is arbitration, selection (with attention), message-out (initiator identifies the LUN), command (initiator sends the CDB), data-in (target sends data to the initiator) *or* data-out (initiator sends data to the target), status (target reports status), and message-in (target reports command completion).



The CDB defines the operation to be performed by the device server. A CDB may have a fixed length of 6,10, 12, or 16 bytes, as shown in the next figure, or a variable length, as shown in the figure after that, of between 12 and 260 bytes. The operation code is the first byte in all CDB formats. The control byte is the last byte in fixed length CDB formats and the second byte in variable length CDB formats.

Fixed length CDB format: (see SAM-2)

Bit Byte	7	6	5	4	3	2	1	0
0	GROUP CODE			COMMAND CODE				
1	Command specific parameters							
n-1	Command specific parameters							
n	Vendor specific	Reserved			NACA	Obsolete	LINK	

Variable length CDB format: (see SPC-2)

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (7Fh)							
1	CONTROL							
2	Reserved							
3	Reserved							
4	Reserved							
5	Reserved							
6	Reserved							
7	ADDITIONAL CDB LENGTH (n-7)							
8	(MSB)	SERVICE ACTION						(LSB)
9								
10	Service action specific fields							
n								

The INQUIRY command (operation code 12h), processed by all SCSI device servers, requests standard device information, vital product data (VPD), or information about which commands are supported by the device server.

Standard INQUIRY data includes ASCII data identifying the manufacturer (8 bytes), the product (16 bytes), the product revision level (4 bytes), and hex data identifying up to eight standards to which the device claims conformance (in 2 byte fields, such as 0900h for FCP-2, 0940h for SRP, and 0960h for iSCSI). Note that T10 maintains a list of SCSI Vendor ID assignments at <http://www.t10.org/lists/vid-alph.htm>

The application client requests VPD data by setting the EVPD bit to one and specifying the page code of the desired VPD data. The supported VPD pages (code 00h) and device identification page (code 83h) are mandatory. Optionally, the INQUIRY command may support the unit serial number page (code 80h), the target operating definition page (code 82h), the FRU information page (codes 01h-7Fh), and vendor-specific pages (codes C0h-FFh). The unit serial number page returns a vendor-assigned serial number for a target or logical unit.

The application client requests command support data by setting the CMDDDT bit to one and specifying the SCSI operation code of the desired CDB. Information returned indicates if the requested operation is supported and if so, if it is implemented in conformance with a SCSI standard or in a vendor-specific manner. However, as this does not support VL CDBs or service actions, the REPORT SUPPORTED OPERATION CODES command (operation code A3h) has been proposed as a more comprehensive method to determine which commands a logical unit supports.

The REPORT LUNS command (operation code A0h) requests a list of the logical unit numbers of all logical units connected to the device. This inventory may also include logical unit numbers for vendor-specific logical units. A device supporting multiple LUN addresses must support a REPORT LUNS command addressed to LUN 0.

Command support is optional for a device having a single logical unit or by logical units other than zero.

SCSI addressing information:

Name is unique within a specified context and does not change. Also called a world-wide identification (**WWID**).

SCSI Identifier represents either an initiator port or a target port identifier. Also called device identifier and port identifier. Note that other standards define the value of SCSI identifiers. For example, SPI-2 defines target identifiers to be in the range 0-7, 0-15, and 0-31.

Task Identifier, assigned by a device server, is an initiator identifier, a logical unit number, and if the task is tagged, a tag. A task identifier in use is unique if one or more of its components is unique.

Tag contains up to 64 bits, assigned by an initiator to ensure the unique identity of a tagged task within a single task set. A tagged task also includes one of the task attributes (simple, ordered, head of queue, or auto contingent allegiance) that allows an initiator to specify tagged task processing relationships.

Nexus is the relationship between a SCSI initiator port, a SCSI target port, optionally a logical unit, and optionally a task.

Identification Descriptor identifies a device or port with the same descriptor when accessed through any path. The INQUIRY device identification page (code 83h) is used to retrieve identification descriptors. Logical units may have more than one identification descriptor if several identifier types are supported. Identifier types may be vendor-specific (not guaranteed to be globally unique), use an 8 byte Vendor ID prefix (vendor guarantees uniqueness), comply with the IEEE 64-bit Global Identifier standard (EUI-64), or be a FC-FS Name Identifier (WWN). Port descriptors are 4 byte binary relative identifiers from 1h ("port A") to 7FFFFFFh ("port 2 147 483 647").

Logical Unit Identifier (LUID) uniquely identifies a logical unit in a SCSI domain.

Logical Unit Number (LUN) is a 64 bit structure that allows up to 4 levels of devices to be addressed under a single target, as shown in the following figure. Each level uses 2 bytes to address the devices on that level. The high order 2 bits indicate the address method (00b = peripheral device, 10b = logical unit, 01b = device type specific) and the remaining 14 bits are address method specific. When a command is sent to the target, the LUN is adjusted to create a new LUN by shifting each 2 byte addressing field up a level, filling in with zeros. Targets that contain 256 or fewer logical units use a single level LUN structure, meaning all fields are 0b except byte 1.

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)	FIRST LEVEL ADDRESSING						(LSB)
1		SECOND LEVEL ADDRESSING						(LSB)
2	(MSB)	THIRD LEVEL ADDRESSING						(LSB)
3		FOURTH LEVEL ADDRESSING						(LSB)
4	(MSB)							(LSB)
5								(LSB)
6	(MSB)							(LSB)
7								(LSB)

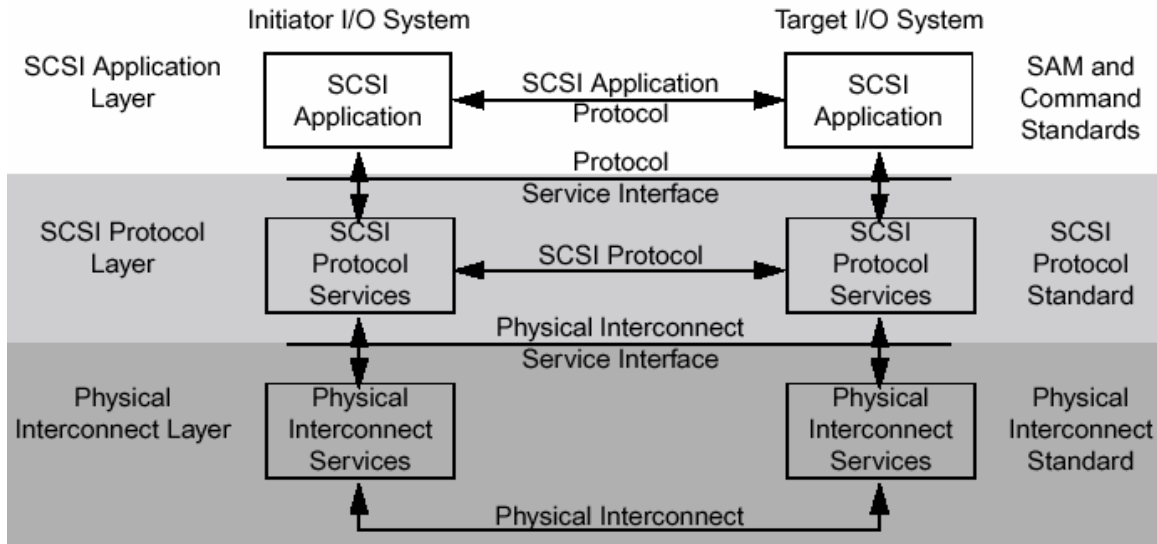
The peripheral device address method, shown in the next figure, relays commands to LUN 0 of the target identified by the 1 byte target/LUN field. LUN 0, used to determine information about a target and its logical units, always uses this address method. A bus identifier of zero indicates a logical unit at the current level. A bus identifier from 1 to 63 represents a bus that physically connects a group of devices to the current level device. If the range of possible target identifiers is too large to fit in 1 byte, the target/LUN field may contain a mapped representation of the target identifier.

Bit Byte	7	6	5	4	3	2	1	0
n-1	0	0	BUS IDENTIFIER					
n	TARGET/LUN							

The logical unit address method, shown in the following figure, relays commands to the LUN within the target located on bus number as indicated. If the range of possible target identifiers is too large to fit in 5 bits, the target field may contain a mapped representation of the target identifier.

Bit Byte	7	6	5	4	3	2	1	0
n-1	1	0	TARGET					
n	BUS NUMBER				LUN			

SCSI defines an application layer, a transport protocol layer, and a physical interconnect layer as shown in the next figure. The subsystems that make up the protocol and interconnect layers are collectively referred to as the service delivery subsystem. A client sends a request to a server that returns a response at the application layer. The application layer uses the services of the protocol layer to implement this transaction. At the client-side, the upper layer protocol (ULP) uses a *Protocol service request* to invoke a service provided by the lower layer protocol (LLP). At the server-side, a *Protocol service indication* from the LLP to a ULP signals that an asynchronous event has occurred. The server-side ULP calls the LLP to respond to the indication using a *Protocol service response*. At the client-side, a *Protocol service confirmation* sent from the LLP to the ULP signals that the request has completed.



Example SCSI Application Protocols are defined in the SPC-3 shared command set and device type specific command sets such as SBC-2 (disk), SCC-2 (RAID controller), SSC-2 (tape), and SES (enclosure).

Example SCSI Transport Protocols are defined in FCP-2, SSA-S3P, SRP, and iSCSI. Example SCSI Physical Interconnects are defined in FC-FS, FC-AL-2, SPI-4 (parallel SCSI), SSA-PH-2, VI, InfiniBand, and Gigabit Ethernet.

Fibre Channel Protocol (FCP-2)

FCP (an FC-4) maps SCSI (a ULP) I/O operations defined by SAM-2 into a Fibre Channel Exchange using FC-FS services. A layered communication model similar to the SCSI request, indication, confirmation, response model is provided by FC-FS. The FC-4 generates a *FC_FS_SEQUENCE.request* to define a transfer of one or more ULP data blocks. The source FC-FS may return a *FC_FS_SEQUENCE_TAG.indication*, then segments the data blocks into frames and transmits the frames as a single Sequence. After receiving the frames, the destination FC-FS reassembles the ULP data blocks, issues acknowledgements as appropriate, and generates a *FC_FS_SEQUENCE.indication* to define the transfer of the completed Sequence to the appropriate destination FC-4. After Sequence delivery has been completed, the source FC-FS may issue *FC_FS_SEQUENCE.confirmation* to indicate success or failure of the request.

FCP is described in terms of Information Units (IUs) and Exchanges generated by a pair of FCP_Ports. Each IU is contained in a single Sequence and each Sequence may only carry a single IU. Up to 65,35 Exchanges and up to 256 active Sequences may be open simultaneously between an initiator and a target FCP_Port. However, some Exchange IDs and at least one extra Sequence ID should always be available for task management.

IUs sent to targets:

- FCP_CMND carries either a SCSI Command or a task management request.
- FCP_CONF is used to confirm the receipt of a FCP_RSP.

IUs sent to initiators:

- FCP_XFER_RDY indicates that the target is prepared to receive part or all of the data for a write command.
- FCP_RSP provides completion information for FCP I/O operations.

IUs sent to both initiators and targets:

- FCP_DATA is used to transfer data in the same Exchange that sent the FCP_CMND requesting the transfer. If more than one IU is used to transfer the data, the relative offset is used to ensure that the SCSI data is reassembled in the proper order.

FCP Information Units are depicted in the following table.

Field	Bits	Value	Notes
<i>FCP_CMND (28 bytes + lengths of additional CDB and Data Buffer Size)</i>			
FCP_LUN	64		SCSI command or task management request LU
Command Reference Number (CRN)	8		Confirms receipt and ordering of commands
Reserved	5		
Task Attribute	3		000b=Simple, 001b=Head of Queue, 010b=Ordered, 100b=ACA, 101b=Untagged
obsolete	1		<i>Task Management Flag</i> byte indicates task management requests using a new exchange:
CLEAR ACA	1		
TARGET RESET	1		
LOGICAL UNIT RESET	1		
Reserved	1		
CLEAR TASK SET	1		
ABORT TASK SET	1		
reserved	1		
Additional CDB Length	6	x	x is a multiple of 4 byte words
RDDATA	1		Initiator expects <i>FCP_DATA</i> IUs for SCSI Read
WRDATA	1		Initiator expects <i>FCP_DATA</i> IUs for SCSI Write
FCP_CDB	128		CDB to be sent to addressed LU
Additional FCP_CDB	4*x*8		Used when CDB is greater than 16 bytes
FCP_DL (Data Buffer Size)	32		Maximum bytes to be transferred to/from buffer
<i>FCP_XFER_RDY (12 bytes)</i>			
FCP_DATA_RO	32		SAM-2 application client buffer offset which may be used by target to request out of order write data if allowed by FMDP disc-recon page
FCP_BURST_LEN	32		SAM-2 data delivery request byte count used to request the initiator to send IU of this length
Reserved	32		
<i>FCP_DATA (transmitted in the same exchange that sent the FCP_CMND requesting the transfer)</i>			
<i>FCP_RSP (24 bytes plus length of sense and response data)</i>			
Reserved	83		Completion information for FCP I/O operations
FCP_CONF_REQ	1		Initiator to send <i>FCP_CONF</i> to confirm <i>FCP_RSP</i>
FCP_RESID_UNDER	1		<i>FCP_RESID</i> expected bytes were not transferred
FCP_RESID_OVER	1		<i>FCP_DL</i> too short to hold <i>FCP_RESID</i> bytes
FCP_SNS_LEN_VALID	1		Examine <i>FCP_SNS_INFO</i> for possible error
FCP_RSP_LEN_VALID	1		Examine <i>FCP_RSP_INFO</i> for possible error
SCSI Status Code	8		SAM-2 Status code (if <i>FCP_RSP_LEN_VALID</i> =0b)
FCP_RESID	32		Number of bytes that were not transferred
FCP_SNS_LEN	32	n	Number of bytes in <i>FCP_SNS_INFO</i> field
FCP_RSP_LEN	32		Number of bytes in <i>FCP_RSP_INFO</i> field
FCP_RSP_INFO	64		Next 8 bytes is <i>FCP_RSP_INFO</i> field if valid
Reserved	24		
RSP_CODE	8		Protocol failure information: 00h=Task Management function complete, 04h=Task Management function rejected, 05h=Task Management function failed, 01h= <i>FCP_DATA</i> length different than <i>FCP_BURST_LEN</i> , 02h= <i>FCP_CMND</i> fields invalid, 03h= <i>FCP_DATA</i> parameter does not match <i>FCP_DATA_RO</i>
Reserved	32		
FCP_SNS_INFO	n		SPC-2 autosense data
<i>FCP_CONF (no payload - confirms receipt of FCP_RSP if supported and requested)</i>			

The address of each FCP_Port is defined by its address identifier. Each FCP I/O operation is identified by the FCP I/O operation's fully qualified exchange identifier (FQXID). The FQXID is composed of the initiator address identifier, the target address identifier, the OX_ID and the RX_ID. Addressability of logical units uses the logical unit number provided in the FCP_CMND IU. Subsequent identification of the FCP I/O operation and the Exchange which carries the protocol interactions for the FCP I/O operation uses the FQXID. FCP devices do not use the Process_Associator. The target uses the OX_ID, and, if it has been assigned, the RX_ID to perform error

recovery and task management functions. The task retry identifier is used as a supplemental task identifier if task retry identification is supported and enabled.

An initiator completes a Process Login (PRLI) with a target before exchanging FCP IUs to identify the capabilities that the Originator FCP_Port expects to use with the Responder FCP_Port and to determine the capabilities of the Responder. Each target has knowledge of the Port Name of each initiator through the FC login process. If a target receives a PRLI or an N_Port Login (PLOGI) from an initiator FCP Port with a previously known WWN but with a changed initiator identifier, the device server shall assign the new initiator identifier to the existing registration and reservation to the initiator port having the same WWN. Each logical unit shall be able to present a WWN through the INQUIRY command vital product data device identification page.