draft proposed
American National Standard

X3.269-199X

Revision 012
December 4, 1995

Information Systems -
dpANS Fibre Channel Protocol for SCSI

This is a draft proposed American National Standard of Accredited Standards Committee X3. As such this is not a completed standard. The X3T10 Technical Committee may modify this document as a result of comments received during public review and its approval as a standard.

ABSTRACT

This standard describes the frame format and protocol definitions required to transfer commands and data between a SCSI (Small Computer System Interface) Initiator and Target using the Fibre Channel Physical and Signaling Interface.

Technical editor:

Robert Snively
Sun Microsystems, UMPK 12-204
2550 Garcia Ave.
Mountain View, CA 94043-1100
(415) 786-6694
Email: bob.snively@eng.sun.com

Fibre Channel Leader:

I. Dal Allan
ENDL
14426 Black Walnut Court
Saratoga, CA 95070
(408) 867-6630
Email:0002501752@mcimail.com

ISO
LOGO

IEC
LOGO

Reference number
ISO/IEC xxxxx: 199x
ANSI X3.xxx: 199x

POINTS OF CONTACT:

X3T10 Chair	X3T10 Vice Chair
John Lohmeyer Symbios Logic 1635 Aeroplaza Dr. Colorado Springs, CO 80916 (719) 573-3362 Email: john.lohmeyer@symbios.com	Larry Lamers Adaptec, Inc. 691 South Milpitas Blvd. Milpitas, CA 95035 (408) 957-7817 Email: ljlamers@aol.com

X3 Secretariat

Lynn Barra
Administrator Standards Processing
X3 Secretariat
1250 Eye Street, NW Suite 200
Washington, DC 20005

Telephone: 202-626-5738
Facsimile: 202-638-4922

SCSI Reflector

Internet address for subscription to SCSI reflector:	scsiadm@symbios.com
Internet address for distribution via SCSI reflectors	scsi@symbios.com

SCSI Bulletin Board

719-574-0424

Document Distribution

Global Engineering
15 Inverness Way East
Englewood, CO 80112-5704

Telephone: 303-792-2181 or
800-854-7179
Facsimile: 303-792-2192

PATENT STATEMENT

CAUTION: The developers of this standard have requested that holder's of patents that may be required for the implementation of the standard, disclose such patents to the publisher. However, neither the developers nor the publisher have undertaken a patent search in order to identify which, if any, patents may apply to this standard.

As of the date of publication of this standard and following calls for the identification of patents that may be required for the implementation of the standard, no such claims have been made. No further patent search is conducted by the developer or the publisher in respect to any standard it processes. No representation is made or implied that licenses are not required to avoid infringement in the use of this standard.

Contents

Tables

Figures

Annexes

Foreword (This foreword is not part of American National Standard X3.269-199x.)

The Fibre Channel Protocol for SCSI (FCP) defines a Fibre Channel mapping layer (FC-4) that uses the services defined by ANS X3.230-199X, Fibre Channel - Physical and Signaling Interface (FC-PH) to transmit SCSI command, data, and status information between a SCSI initiator and a SCSI target. The use of the standard enables the transmission of standard SCSI command formats, the transmission of standard SCSI data and parameter strings, and the receipt of SCSI status and sense information across the Fibre Channel using only the standard Fibre Channel frame and sequence formats. The FCP will operate with Fibre Channel Classes of Service 1, 2, and 3 and will operate across Fibre Channel fabrics and arbitrated loops.

Subcommittee X3T10 on Lower-Level Interfaces, which developed this standard, had the following members:

John B. Lohmeyer, Chair

Lawrence J. Lamers, Vice-Chair

Ralph Weber, Secretary

Jason Albanus
I. Dal Allan
Paul D. Aloisi
Bruce Anderson
Geoffrey Barton
Robert Bellino
Charles Brill
Peter Brown
Michael Bryan
Joe Chen
Joe Dambach
Jan V. Dedek
Peter Dougherty
Stephen G. Finch
Bill Galloway
Louis Grantham
Norm Harris
Edward Haske
Stephen F. Heil
Peter Johansson
Gerry Johnsen
Skip Jones
Ray Kallio
Sam Karunanithi
Nick Ladas
Edward Lappin
David Lawson
Robert Liu
Bob Masterson
David McFadden
James McGrath
Patrick Mercer
Charles Monia
Ian Morrell
John Moy
Erich Oetting
Alan R. Olson
Dennis Pak
Duncan Penman
George Penokie
Doug Piper
Donna Pope
Ron Roberts
Jeff Rosa
Robbie Shergill
Scott Smyers
Robert N. Snively
Jeff Stai
Gary R. Stephens
Clifford E. Strang, Jr.
Dean Wallace
Harvey Waltersdorf
Neil T. Wanamaker
David Wang
Jeffrey L. Williams
Mark Woithe
Ezra Alcudia (Alt.)
Michael Alexenko (Alt.)
David Andreatta (Alt.)
Tak Asami (Alt.)
Akram Atallah (Alt.)
Paul Boulay (Alt.)
Gary Brandvold (Alt.)
Ken Bush (Alt.)
John Cannon (Alt.)
Kurt Chan (Alt.)
Shufan Chan (Alt.)
Ting Le Chan (Alt.)
William Clemmey (Alt.)
Dan Colegrove (Alt.)
Roger Cummings
William Dallas (Alt.)
Kim Davies (Alt.)
Varouj Der-Hacopian (Alt.)
Timothy Feldman (Alt.)
Edward A. Gardner (Alt.)
John Geldman (Alt.)
Chuck Grant (Alt.)
Douglas Hagerman (Alt.)
Kenneth J. Hallam (Alt.)
William Harn (Alt.)
Tom Hanan (Alt.)
Rick Heidick (Alt.)
Mike Hetzel (Alt.)
Gerald Houlder (Alt.)
Brian Johnson (Alt.)
Mark Jordan (Alt.)
Richard Kalish (Alt.)
Greg Kapraun (Alt.)
Prabhas Kejriwal (Alt.)
Dan Kennedy (Alt.)
Thomas J. Kulesza (Alt.)
Dennis Lang (Alt.)
Joe Lawlor (Alt.)
Gerald Marazas (Alt.)
John Masiewicz (Alt.)
Pete McLean (Alt.)
Daniel E. Moczarny (Alt.)
E. J. Mondor (Alt.)

Pete Tobias (Alt.)
Harvey Toyama (Alt.)
Adrienne Turenne (Alt.)
Joel Urban (Alt.)
Joseph Wach (Alt.)
Roger Wang (Alt.)
Dave Weber (Alt.)
Bob Whiteman (Alt.)
Michael Wingard (Alt.)
Kurt Witte (Alt.)
Devon Worrell (Alt.)
Charles I. Yang (Alt.)
Mike Yokoyama (Alt.)
Vit Novak (Alt.)
Kevin R. Pokorney (Alt.)
Gary Porter (Alt.)
Steven Ramberg (Alt.)
Mark Raymond (Alt.)
Simone Sadri (Alt.)
John P. Scheible (Alt.)
Stan Sharp (Alt.)
Michael Smith (Alt.)
Allen Spalding (Alt.)
Arlan P. Stone (Alt.)
Joe Stoupa (Alt.)
George Su (Alt.)
Nicos Syrimis (Alt.)
Rich Taborek (Alt.)

Subcommittee X3T11 on Device Level Interfaces, which contributed a great deal of analysis, review, and content to this standard, had the following members:

Roger Cummings, Chair

Carl Zeitler, Vice-Chair

Neil Wanamaker, Secretary

J. Albano
D. Allan
R. Anderson
T. Anderson
K. Annamalai
C. Barron
B. Beach
A. Bedi
B. Bellino
H. Brandt
C. Brill
R. Brown
M. Carboni
B. Carlson
K. Chan
K. Chennappan
B. Cook
J. Connell
B. Cornelius
S. Dean
D. Deel
J. Dudgeon
R. Dugan
S. Erler
G. Frazier
G. Frederickson
E. Frymoyer
B. Gallagher
S. Gillett
E. Grivna
J. Guedj
T. Kaminski
J. Kerr
R. Knerr
F. Koperda
B. Kopsaftis
O. Kornblum
G. Koskowich
J. Kubinec
P. Lambert
P. Levin
R. Liebow
J. Luttrull
S. Magee
K. Malavalli
C. Martin
R. Masterson
J. Mathis
T. McClendon
G. Milligan
G. Michael
M. Miller
B. Morris
J. Morris
D. Morrissey
R. Niemi
G. Nordstrom
M. O'Donnell
K. Ocheltree
T. Palkert
E. Parish
N. Perdue
P. Rupert
T. Russell
M. Sachs
W. Sanderson
P. Savage
J. Scherfey
P. Schreck
J. Schuessler
P. Scott
F. Shott
K. Sivakumar
L. Sloan
J. Smith
R. Snively
R. Soderstrom
D. Somes
J. Soriano
G. Stephens
R. Southard
T. Sprenkle
G. Stager
D. Stigliani
A. Stone
B. Stoevhase
T. Szostak
R. Taborek
R. Thomsen
H. Todd
D. Tolmie
C. Towndrow
H. Truestedt

D. Guss
D. Gustafson
M. Halvorson
D. Hansen
K. Hardwick
M. Hartmann
M. Henske
D. Hepner
J. Huber
J. Hughes
R. Johnson
G. Phillips
R. Phillips
J. Radcliffe
P. Ramsey
G. Rara
E. Reid
J. Renwick
W. Rickard
W. Roiger
R. Ronald
G. Rouse
E. Vanderbeck
S. van Doorn
L. Veil
M. Vogel
P. Walford
Y. Wang
L. Whitfield
A. Widmer
J. Young
S. Zanowick

Suggestions for improvement, requests for interpretation, addenda, or defect reports are welcome. They should be sent to the X3 Secretariat, Information Technology Industrial Council (ITI), 1250 Eye Street, NW, Suite 200, Washington, DC 20005.

This standard was processed and approved for submittal to ANSI by Accredited Standards Committee on Information Technology, X3. Committee approval of the standard does not necessarily imply that all committee members voted for its approval. At the time it approved this standard, the X3 Committee had the following members:

name, Chair

name, Vice-Chair

name, Secretary

name
name
name
name
name
name
name
name
name
name
name
name
name
name
name
name
name
name

Introduction

The Small Computer System Interface (SCSI) command set is widely used and applicable to a wide variety of device types. The transmission of SCSI command set information across Fibre Channel links allows the large body of SCSI application and driver software to be successfully used in the high performance Fibre Channel environment.

This document describes a protocol for using Fibre Channel FC-PH Exchanges and Information Units to implement the SCSI Fibre Channel Protocol (FCP). The Fibre Channel is a high speed serial architecture that allows either optical or electrical connections at data rates from 133 Mbits up to 4 Gbits per second. The topologies supported by Fibre Channel include point-to-point, fabric switched, and arbitrated loop. All Fibre Channel connections use the same standard frame format and standard hierarchy of transmission units to transmit the Information Units that carry SCSI information.

The Fibre Channel Protocol for SCSI standard is divided into 8 major clauses:

Clause 1 is the scope.

Clause 2 enumerates the normative references that apply to this standard.

Clause 3 describes the definitions, abbreviations, and conventions used in this standard.

Clause 4 describes an overview of the Fibre Channel Protocol.

Clause 5 describes the information units used to transfer SCSI commands, data, and status across a Fibre Channel connection.

Clause 6 describes the FCP Basic Link Services and Extended Link Services used by the FCP.

Clause 7 describes the format of the individual information units used to transfer SCSI commands, data and status across a Fibre Channel connection.

The Fibre Channel Protocol for SCSI standard has three additional annexes.

Annex A is a normative annex that provides a description of the Process Login and Process Logout capabilities. These functions are expected to be defined in a future extension of the Fibre Channel standard.

Annex B is an informative annex that provides examples of the Fibre Channel Protocol operations.

Annex C is an informative annex that provides an example of how the address space of SCSI logical units can be organized.

Annex D is an informative annex that lists standards that should be of interest to the reader, but are not essential to complete the requirements of this standard.

The Fibre Channel Protocol for SCSI standard is part of the SCSI-3 family of standards developed by X3T10 to facilitate the use of the SCSI command sets for many different types of devices across many different types of physical interconnects. The master architectural document of the family of standards is ANSI X3.270-199X, *Information Technology - SCSI-3 Architecture Model (SAM)* The SAM document contains a guide to other documents in the SCSI-3

Information Technology-
Fibre Channel Protocol for SCSI

1 Scope

This standard defines the SCSI Fibre Channel Protocol (FCP). The FCP is a mapping protocol (FC-4) for applying the SCSI command set to the Fibre Channel. The FCP defines the Fibre Channel information units in accordance with the SCSI Architecture Model (X3.270-199X). The FCP additionally defines how the Fibre Channel services are used to perform the services defined by the SCSI Architecture Model.

2 Normative references

The following standard contains provisions which, through reference in the FCP, constitute provisions of this standard. At the time of publication, the edition indicated was valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standard listed below. Members of IEC and ISO maintain registers of currently valid International Standards, and ANSI maintains registers for American National Standards.

ANSI X3.131-1994, *Information Systems - Small Computer System Interface - 2 (SCSI-2)*

ANSI X3.230-1994, *Information Technology - Fibre Channel - Physical and Signaling Interface (FC-PH)*

ANSI X3.270-199X, *Information Technology - SCSI-3 Architecture Model (SAM)*

Reference is made to these documents by their standard designations. See Annex D for a bibliography of additional references which are not yet standards.

3 Definitions and abbreviations

Definitions, conventions, abbreviations, acronyms and symbols applicable to this standard are provided, unless they are identical to that described in any referenced standard, in which case they are included by reference. Some definitions from the glossary or body of other standards are included here for easy reference.

3.1 Definitions

3.1.1 address identifier: An address value used to identify source (S_ID) or destination (D_ID) of a frame. [X3.230-199X]

3.1.2 application client: An object that is the source of SCSI commands. [X3.270-199X]

3.1.3 application client buffer offset: Offset in bytes from the start or base address of the application client's data buffer to the location for the transfer of the first byte of a data delivery service request. [X3.270-199X]

3.1.4 base address: The address of the lowest address byte to be transferred to or from an application client buffer.

3.1.5 command: A request describing a unit of work to be performed by a device server. [X3.270-199X]

3.1.6 command byte count: Upper limit on the extent of the data to be transferred by the SCSI command. [X3.270-199X]

3.1.7 command descriptor block: A structure up to 16 bytes in length used to communicate a command from an application client to a device server. [X3.270-199X]

3.1.8 data in delivery service: A confirmed service used by the device server to request the transfer of data to the application client. [X3.270-199X]

3.1.9 data out delivery service: A confirmed service used by the device server to request the transfer of data from the application client. [X3.270-199X]

3.1.10 Destination_Identifier: The address identifier used to indicate the targeted destination of the transmitted frame. [X3.230-199X]

3.1.11 device server: An object within the logical unit which executes SCSI tasks and enforces the rules for task management. [X3.270-199X]

3.1.12 Execute Command service: A peer-to-peer, confirmed service requested by the application client to perform a SCSI Command. [X3.270-199X]

3.1.13 FCP I/O operation: An unlinked SCSI command, a series of linked SCSI commands, or a task management function. [X3.270-199X]

3.1.14 FCP_Port: An N_Port or NL_Port that supports the SCSI Fibre Channel Protocol.

3.1.15 fully qualified exchange identifier: A token used to uniquely identify a FCP I/O Operation.

3.1.16 information unit: An organized collection of data specified by the FCP to be transferred as a single sequence by the Fibre Channel service interface. [X3.230-199X]

3.1.17 initiator: An SCSI device containing application clients that originate device service requests and task management functions to be processed by a target SCSI device. [X3.270-199X]

3.1.18 initiator identifier: Token by which a target identifies the initiator device. [X3.270-199X]

3.1.19 logical unit: A target resident entity that implements a device model and executes SCSI commands sent by an application client. [X3.270-199X]

3.1.20 logical unit identifier: Identifier used by an initiator to reference the logical unit. [X3.270-199X]

3.1.21 logical unit number: An encoded 64-bit identifier for a logical unit. [X3.270-199X]

3.1.22 NL_Port: An N_Port that contains arbitrated loop functions associated with the Fibre Channel Arbitrated Loop topology. [FC-AL]

3.1.23 N_Port: A hardware entity that supports the FC-PH. It may act as an originator, a responder, or both. [X3.230-199X]

3.1.24 request byte count: Number of bytes to be moved by a data delivery service request. [X3.270-199X]

3.1.25 Source_Identifier: The address identifier used to indicate the source port of the transmitted frame. [X3.230-199X]

3.1.26 SCSI device: A device that originates or services SCSI commands. [X3.270-199X]

3.1.27 status: A single byte returned by the device server to the application client in its response to indicate the completion and completion state of a command. [X3.270-199X]

3.1.28 tag: The initiator-specified component of the task identifier. [X3.270-199X]

3.1.29 target: An SCSI device that receives SCSI commands and directs such commands to one or more logical units for execution. [X3.270-199X]

3.1.30 target identifier: Address of up to 64-bits by which a target is identified. [X3.270-199X]

3.1.31 task: An object within the logical unit representing the work associated with a command or group of linked commands. [X3.270-199X]

3.1.32 task attribute: The queuing specification for a task (SIMPLE, ORDERED, HEAD OF QUEUE, ACA).

[X3.270-199X]

3.1.33 task identifier: The information uniquely identifying a task. [X3.270-199X]

3.1.34 task management function: A peer-to-peer confirmed service provided by a task manager that can be invoked by an application client to affect the execution of one or more tasks. [X3.270-199X]

3.2 Abbreviations

D_ID	Destination_Identifier[X3.230-199X]
FC	Fibre Channel [X3.230-199X]
FC-PH	The architecture specified by the Fibre Channel standard. [X3.230-199X]
FC-4	Fibre Channel Layer 4 mapping layer. [X3.230-199X]
FQXID	fully qualified exchange identifier (See 3.1.15).
ID	identifier
IPA	initial process associator
IU	information unit [X3.230-199X]
SCSI	Small Computer System Interface. Either SCSI-2 or SCSI-3.
SCSI-2	The architecture specified by ANS X3.131-1994.
SCSI-3	The architecture specified by ANS X3.270-199X.
S_ID	Source_Idnetifier[X3.230-199X]
ULP	upper layer protocol [X3.230-199X]

3.3 Editorial conventions

In all of the figures, tables, and text of this document, the most significant bit of a binary quantity is shown on the left side.

In case of any conflict between figure, table, and text, the text takes precedence. Exceptions to this convention are indicated in the appropriate subclauses.

The term “shall” is used to indicate a mandatory rule. If such a rule is not followed, the results are unpredictable unless indicated otherwise.

4 General

4.1 Structure and concepts

The Fibre Channel (FC) is logically a point-to-point serial data channel. The architecture has been designed so that it can be implemented with high performance hardware that requires little real-time software management. The Fibre Channel Physical layer described by X3.230-199X performs those functions required to transfer data from one N_Port to another. The FC-PH layer can be treated as a very powerful delivery service with information grouping and three defined classes of service.

A switching fabric allows communication among more than two N_Ports.

The Fibre Channel Arbitrated Loop is an alternative implementation that uses the FC mechanisms to transfer data between an NL_Port selected by the arbitration process and any of the other NL_Ports on the loop. Once communication is opened between two NL_Ports, standard FC frames are used to provide an FC-PH compliant delivery service.

An FC-4 mapping layer uses the services provided by FC-PH to execute the steps required to perform the functions defined by the FC-4. The protocol is described in terms of the stream of FC IUs and exchanges generated by a pair of N_Ports or NL_Ports that support the FC-4. In the FCP document, N_Ports and NL_Ports capable of supporting FCP transactions are collectively referred to as FCP_Ports.

The detailed implementation that supports that stream is not defined, although originator and responder FCP_Ports are assumed to have a common service interface for use by all FC-4s that is similar in characteristics to the service interface defined in annex S of X3.230-199X. The requirements for the service interface for the SCSI are contained in X3.270-199X. The SCSI Common Access Method [CAM] is one example of a service interface that fulfills the requirements specified in X3.270-199X.

Four kinds of functional management are defined by FCP.

Device Management

Task Management

Process login/logout Management

Link Management

The FCP device and task management protocols define the mapping of the SCSI functions defined in X3.270-199X to the FC-PH. The FCP is based on a two-level paradigm. The I/O Operation defined by X3.270-199X is mapped into an exchange. The request and response primitives of an I/O Operation are mapped into information units. Link control is performed by standard FC-PH protocols. This is shown in table 1.

Table 1 - SCSI and FCP functions

SCSI function	FCP equivalent
I/O Operation	Exchange
Request/Response Primitives	Sequence
Command service request	Unsolicited command IU (FCP_CMND)
Data delivery request	Data descriptor IU (FCP_XFER_RDY)
Data delivery action	Solicited data IU (FCP_DATA)
Command service response	Command status IU (FCP_ RSP)

The number of exchanges that may simultaneously be open between an initiator FCP_Port and a target FCP_Port is defined by the FC-PH implementation. The architectural limit for this value is 65535. The maximum number of active sequences that can simultaneously be open between an initiator FCP_Port and a target FCP_Port is defined by the FC-PH Sequence_ID as 256. To allow task management exchanges to be originated, a certain number of extra exchange IDs and at least one extra Sequence_ID should always be available.

4.2 Device management

An application client begins a FCP I/O Operation when it provides to the FCP a request for an Execute command service. A single request or a list of linked requests may be presented to the software interface of the FCP. Each request contains all the information necessary for the execution of one SCSI command, including the local storage address and characteristics of data to be transferred by the command. The FCP then performs the following actions using FC-PH services to perform the SCSI command.

The FCP_Port that is the SCSI initiator for the command starts an exchange by sending an unsolicited command IU containing the FCP_CMND payload, including some command control flags, addressing information, and the SCSI command descriptor block (CDB). The FCP_CMND payload is the Execute Command service request and starts the FCP I/O Operation. The exchange that is started is identified by its fully qualified exchange identifier (FQXID) during the remainder of the FCP I/O Operation and is used only for the IUs associated with that FCP I/O Operation.

When the device server for the command has completed the interpretation of the command, has determined that data transfer is required, and is prepared to request the data delivery service, it sends a data descriptor IU containing the FCP_XFER_RDY payload to the initiator to indicate which portion of the data is to be transferred. If the SCSI command described a write operation, the FCP_Port that is the SCSI initiator then transmits a solicited data IU to the target containing the FCP_DATA payload requested by the FCP_XFER_RDY payload. If the SCSI command describes a read operation, the FCP_Port that is the SCSI target transmits a solicited data IU to the initiator containing the FCP_DATA payload it had described in the FCP_XFER_RDY payload. Data delivery requests containing FCP_XFER_RDY and FCP_DATA payloads continue until all data described by the SCSI command is transferred. Exactly one FCP_DATA IU follows each FCP_XFER_RDY IU. The transmission of the FCP_XFER_RDY IUs may be disabled for those systems having other mechanisms for controlling the data transfer.

After all the data has been transferred, the device server transmits the Execute Command service response by requesting the transmission of an IU containing the FCP_RSP payload. That payload contains the SCSI status and, if an unusual condition has been detected, the SCSI REQUEST SENSE information and the FCP response information describing the condition. The command status IU terminates the command. The SCSI logical unit determines whether additional commands will be performed in the FCP I/O Operation. If this is the last or only command executed in the FCP I/O Operation, the FCP I/O Operation and the exchange are terminated.

When the command is completed, returned information is used to prepare and return the Execute Command service confirmation information to the software that requested the operation. The returned status indicates whether or not the command was successful. The successful completion of the command indicates that the SCSI device performed the desired operations with the transferred data and that the information was successfully transferred to or from the SCSI initiator. A protocol is provided to present error information if a command is not successful.

If the command is linked to another command, the FCP_RSP payload shall contain the proper status indicating that another command will be executed. The target shall present the FCP_RSP in an IU that allows command linking. The initiator shall continue the same exchange with an FCP_CMND IU, beginning the next SCSI command. All SCSI commands linked in the FCP I/O Operation except the last are executed in the manner described above.

The number of FCP I/O Operations that may be active at one time depends on the queuing capabilities of the particular SCSI devices and the number of concurrent exchanges supported by the FCP_Ports.

The FCP takes full advantage of the multiplexing and shared bandwidth capabilities of FC-PH Class 2 operation. Class 1 and intermixed classes of service may also be used to transfer the IUs of FCP I/O Operations. Class 3 service may be used. The error recovery characteristics of Class 3 may require that it be allowed only in certain operating environments to meet reliability and error detection requirements. Fabrics and arbitrated loops may be used to allow access among multiple initiators and targets.

Both FC-PH and SCSI allow the initiator function in any FCP_Port and the target function in any FCP_Port. For FCP I/O Operations between a host and a peripheral subsystem, the host normally takes on the SCSI initiator role and the peripheral subsystem normally takes on the SCSI target role. For host to host or device to device communications, either one of the communicating pair can take on the SCSI initiator role. For Asynchronous Event Notification, the peripheral device takes on the SCSI initiator role to inform the host, in its target role, that an asynchronous event has occurred.

4.3 Task management

An application client requests a task management function when a task or some group of tasks must be aborted or terminated. See “Task management flags, Byte 2” on page 26.

Some SCSI task management functions are mapped into FC-PH link services and others are mapped into control bits in the FCP_CMND IU. Task management functions that use the FCP_CMND IU are transmitted as the first IU in a new exchange, except for TERMINATE TASK which is performed in the same exchange. A task management function ends with an FCP_RSP IU that indicates whether it was correctly accepted. The FCP_CDB field in FCP_CMND IUs that perform task management functions is ignored. The mappings are explained in table 2.

Table 2 - Task management functions, SCSI-3 to FCP

SCSI-3 function	FCP equivalent	Optional/ required
TARGET RESET	FCP_CNTL TARGET RESET	Required
ABORT TASK	FCP recovery abort	Required
ABORT TASK SET	FCP_CNTL ABORT TASK SET	Required
CLEAR TASK SET	FCP_CNTL CLEAR TASK SET	Required
TERMINATE TASK	FCP_CNTL TERMINATE TASK	Optional
CLEAR ACA	FCP_CNTL CLEAR ACA	Required*
* CLEAR ACA is required if Automatic Contingent Allegiance is used.		

The ABORT TASK function and any required FC-PH reinitialization or clean-up functions are performed using FC-PH link management functions.

4.4 Process login/logout

The Process login/logout (PRLI/PRLO) extended link service is optionally used to establish the operating relationships between two FCP_Ports. Implicit PRLI/PRLO parameters may be defined for FCP_Ports.

4.5 Link management

X3.230-199X allows management protocols above the FC-PH interface to perform link data functions. The standard FC-PH primitive sequences, link management protocols, and basic and extended link services are used as required by FCP devices. Implicit login functions are allowed.

5 FCP protocol overview

5.1 FCP addressing and exchange identification

The address of each FCP_Port is defined by its address identifier as described in X3.230-199X. Each FCP I/O Operation is uniquely identified by the address identifier of the initiator, the address identifier of the target, and parameters that uniquely identify the exchange between the initiator and target. The value that identifies the FCP I/O Operation is defined as the fully qualified exchange identifier (FQXID), defined in the following table. The method used to identify FCP I/O Operations internal to the application client and the device server is not defined by this standard.

Addressability of logical units internal to an addressed NL_Port uses the logical unit number provided in the FCP_CMND IU. Subsequent identification of the FCP I/O Operation (exchange) is done by using the FQXID. More than one logical initiator or logical target image may be defined by the originator process associator or the responder process associator, respectively. The process associator does not take part in the FQXID, but does provide an optional additional layer of logical addressability.

The components required to define the FQXID for basic operation and X_ID invalidation for the initiator and target are indicated by an R (Required) in table 3.

Table 3 - Definitions of fully qualified exchange identifier

Condition	D_ID	S_ID	OX_ID	RX_ID	OOA	ROA
Basic operation, initiator identification	R	R	R			
Basic operation, target identification	R	R	R	R		
X_ID invalidation, initiator identification	R	R			R	
X_ID invalidation, target identification	R	R			R	R

Other definitions of FQXID are outside the scope of X3.269-199X.

The target is required to be cognizant of the OX_ID to perform error recovery and task management functions.

5.2 SCSI address format for FCP

Certain third-party SCSI commands and parameters specify a 64-bit field that is defined to access other SCSI devices addressable from that port. These commands include COPY, RESERVE, and several others. The format of the 64-bit field used by FCP_Ports is defined in table 4.

Table 4 - FCP format for third-party FCP_Port addressability

Bit Byte	7	6	5	4	3	2	1	0
0	reserved							PA_VAL
1–3	(MSB)			FCP_Port identifier				(LSB)
4–7	(MSB)			Process associator				(LSB)

5.2.1 PA_VAL

If this bit is set to *1*, the process associator field is valid. If this bit is set to *0*, the process associator field is not valid and may have any value.

5.2.2 FCP_Port identifier

This field defines the D_ID or S_ID that shall be used by the SCSI target for the required third-party addressing.

5.2.3 Process associator

If the PA_VAL bit indicates that this field is valid, the field defines the process associator that shall be used by the SCSI target for the required third-party addressing.

5.3 FCP information units

The information units used by the FCP and their characteristics are shown in table 5 for IUs sent to targets, and in table 6 for IUs sent to initiators. Examples of typical FCP operations using these IUs are included in annex B.

Table 5 - FCP information units sent to targets

IU	SCSI-3 primitive	Data block		F/M/L	SI	M/O
		CAT	Content			
T1	Command / Task Mgmt Rqst	6	FCP_CMND	F	T	M
T2	Command request	6	FCP_CMND	F	H	O
T3	Command request (Linked)	6	FCP_CMND	M	T	O
T4	Command request (Linked)	6	FCP_CMND	M	H	O
T5	reserved					
T6	Data Out action	1	FCP_DATA	M	T	M
T7	Data Out action	1	FCP_DATA	M	H	O
T8	Command Rqst and Data Out	6/1	FCP_CMND + FCP_DATA	F	T	O
T9	Command Rqst and Data Out	6/1	FCP_CMND + FCP_DATA	F	H	O
T10	Command request and Data Out action (Linked)	6/1	FCP_CMND + FCP_DATA	M	T	O
T11	Command request and Data Out action (Linked)	6/1	FCP_CMND + FCP_DATA	M	H	O

Note:

T2, T4, T7, T8, T9, T10, and T11 are only permitted when transfer ready IUs are disabled.
T8, T9, T10, and T11 are only permitted when PLOGI has allowed categories per sequence > 1.
T3, T4, T10, and T11 are only permitted for linked SCSI commands.
T2, T4, T7, T9, and T11 allow optional sequence streaming during Write operations.

Key:

IU	Information unit identifier
CAT	Information category of Device_Data frames carrying the data block (see X3.230-199X subclause 18.2)
CONTENT	Contents (payload) of data block
F/M/L	First/Middle/Last Sequence of exchange (X3.230-199X subclause 18.5)
SI	Sequence Initiative: Held or Transferred (X3.230-199X subclause 18.5)
M/O	Mandatory/Optional Sequence

Table 6 - FCP information units sent to initiators

IU	SCSI-3 primitive	Data block		F/M/L	SI	M/O
		CAT	**Content**			
I1	Data delivery request	5	FCP_XFER_RDY (Write)	M	T	M
I2	Data delivery request	5	FCP_XFER_RDY (Read)	M	H	M
I3	Data In action	1	FCP_DATA	M	H	M
I4	Command/Task Mgmt Response	7	FCP_RSP	L	T	M
I5	Response (Linked)	7	FCP_RSP	M	T	O
I6	Data in action and response	1/7	FCP_DATA + FCP_RSP	L	T	O
I7	Data in action and response (Linked)	1/7	FCP_DATA + FCP_RSP	M	T	O

Note:

I6 and I7 are only permitted when PLOGI has allowed categories per sequence > 1.
I5 and I7 are only permitted for linked SCSI commands.
I2 and I3 allow optional sequence streaming to I3, I4, I5, I6, or I7.

Key:

IU	Information unit identifier
CAT	Information category of Device_Data frames carrying the data block (X3.230-199X subclause 18.2)
CONTENT	Contents (payload) of data block
F/M/L	First/Middle/Last sequence of exchange (X3.230-199X subclause 18.5)
SI	Sequence Initiative: Held or Transferred (X3.230-199X subclause 18.5)
M/O	Mandatory/Optional sequence

5.4 FCP standard formats

The FCP needs only the standard FC-2 services as described in informative annex S of X3.230-199X. Annex S provides an outline of the FC_PH_SEQUENCE.request used by the FCP to provide all the information required to manage the protocol. No additional capabilities are required to transmit the required FCP IUs. Corresponding FC_PH_SEQUENCE.indication information is provided to the sequence recipient to properly type and categorize the received IUs. Class 1 and Class 2 service provides the FC_PH_SEQUENCE.confirmation primitive that can be used for management of the FCP. The use of the FC_PH_SEQUENCE_TAG.indication primitive may be required by the sequence Initiator.

5.5 FC-PH mappings to SCSI-3 functionality

The format of the standard FC-PH header as used by the FCP is defined in table 7.

Table 7 - FCP frame header

Bits Word	31– 24	23–16	15–08	07–00
0	R_CTL	D_ID		
1	reserved	S_ID		
2	TYPE	F_CTL		
3	SEQ_ID	DF_CTL	SEQ_CNT	
4	OX_ID		RX_ID	
5	RLTV_OFF			

All fields use the standard FC-PH definitions. The following explanations of some of the fields provide information about the use of those fields to implement SCSI-3 functionality.

5.5.1 R_CTL

The R_CTL field identifies the frame as part of a FCP operation and identifies the information category. All sequences containing FCP command, data, response, and data descriptor information shall be composed of Device_Data type frames.

The information category associated with each IU is defined in table 5 and table 6.

5.5.2 D_ID

The D_ID identifies the destination of the frame. The D_ID transmitted by the exchange originator is the SCSI-3 target identifier. The D_ID is one segment of the FQXID.

NOTE - X3T11 intends to develop future extensions of the FC that may allow treatment of the D_ID as an alias for multi-N_Port hunt groups, path reconnection groups, and data striping groups.

5.5.3 S_ID

The S_ID identifies the source of the frame. The S_ID transmitted by the exchange originator is the SCSI-3 initiator identifier. The S_ID is one segment of the FQXID.

NOTE - X3T11 intends to develop future extensions of the FC that may allow treatment of the S_ID as an alias for multi-N_Port hunt groups, path reconnection groups, and data striping groups.

5.5.4 TYPE

The TYPE field shall be 0x08 for all frames of SCSI FCP sequences.

5.5.5 F_CTL

The F_CTL field manages the beginning and normal or abnormal termination of sequences and exchanges. The bits and definitions shall be as defined by X3.230-199X.

5.5.6 SEQ_ID

The SEQ_ID identifies each sequence between a particular exchange originator and exchange responder with a unique value as defined by X3.230-199X.

5.5.7 DF_CTL

The DF_CTL indicates any optional headers that may be present. FCP does not require any optional headers.

5.5.8 SEQ_CNT

The SEQ_CNT field indicates the frame order within the sequence as defined by X3.230-199X.

5.5.9 OX_ID

The OX_ID field is the originator identification of the exchange and is an element of the FQXID. The OX_ID shall be assigned and shall have a value other than hexadecimal ‘FFFF’. The value of the OX_ID is the tag defined by X3.270-199X. Since the value of the OX_ID is required by FC-PH to be unique, there is no requirement for an FCP logical unit to check for overlapping commands.

5.5.10 RX_ID

The RX_ID field is the responder identification of the exchange and is an element of the FQXID. The RX_ID shall be assigned by the exchange responder and may have any value, including the unassigned value of hexadecimal ‘FFFF’. The RX_ID allows the exchange responder to optionally create its own value for the tag defined by X3.270-199X.

5.5.11 RLTV_OFF

The RLTV_OFF field indicates the relative displacement of the first byte of each frame’s payload with reference to the base address of the information category. For the solicited data category (FCP_DATA IUs), the relative offset is the SCSI-3 application client buffer offset and the base address is the beginning address of the application client’s buffer as described by X3.270-199X. For all other information categories, the base address is *0* and refers to the first byte of the IU. The RLTV_OFF is not required if both FCP_Ports can unambiguously reassemble the transmitted IUs using other FC-PH information and the FCP_XFER_RDY IU.

6 FCP link-level protocol

The FCP link-level protocol includes the basic link services and extended link services defined by X3.230-199X. The protocol also includes the process login and process logout extended link services defined below.

Link-level protocols are used to configure the FC environment, including the establishment of configuration information and address information. Devices introduced into a configuration or modifications in the addressing or routing of the configuration may require new login procedures.

6.1 Overview of process login/logout

A new extended link service command of Process login (PRLI) is defined in Annex A. The command uses standard extended link service behavior as defined in X3.230-199X. The PRLI (LS_Command code = “20” hexadecimal) allows for one or more processes at one FCP_Port to be related to corresponding processes at another FCP_Port as an image pair. In addition, the PRLI allows one or more FC-4 capabilities to be announced by the initiating FCP_Port to the recipient FCP_Port. The recipient FCP_Port indicates its acceptance or rejection of the modes of operation in its response to the PRLI request. The PRLI is optional, since implicit login can be established by configuration conventions outside the scope of this standard.

The effect of the creation of image pairs is to create one or more virtual initiators or virtual targets behind each FCP_Port. As an example, an FCP_Port can identify itself to another FCP_Port as having one or more logically separate SCSI FCP initiators, one or more logically separate SCSI FCP targets, and a number of logically separate processes performing other FC-4 mappings. The FCP_Port receiving the PRLI can reject the command, indicating that it cannot support the required functions, or accept the command and commit to the binding of two processes into an image pair.

The FC-4 specific service parameters for FCP are defined in the following subclauses.

The PRLI extended link service command has a complementary process logout (PRLO, LS_Command code = “21” hexadecimal) described in 6.3.

Process login can establish either an Informative or an Exclusive communication between communicating processes.

> Informative communication exchanges information between communicating processes or FCP_Ports to enable the negotiation or the exchange of service parameters.
>
> Exclusive communication explicitly establishes a binding between a process in one N_Port and a process in another N_Port. The binding does not allow any communication types or paths other than those established by the Process login.

The service parameters exchanged between two processes may be either requirements or capabilities.

> Capabilities indicate those FC-4 service parameters that describe the role and state of the node in the FC-4. Such capabilities include channel or device (FC-SB), initiator or target (FCP), and similar values.
>
> Requirements indicate those FC-4 service parameters that must be agreed upon by both nodes for proper operation of an FC-4. Requirements include values such as the parameters controlling the FCP IUs that must be used.

6.2 Process login (PRLI)

The PRLI request is transmitted from an originator FCP_Port to a responder FCP_Port to identify to the destination the capabilities and requirements that the originator FCP_Port expects to use with the responder FCP_Port and to determine the capabilities and requirements of the responder. The PRLI allows more than one service parameter page to be included in each command, so that a single command can create more than one image pair for multiple FC-4s at the same time.

The standard PRLI accept (ACC) is returned to the originator FCP_Port to indicate that the responder FCP_Port accepts the capabilities and agrees to the requirements of the originator according to the rules below.

A link service reject (LS_RJT) indicates that the PRLI request is not accepted.

The PRLI common service parameters and accept response codes are defined in annex A. FC-4 service parameters for mappings other than FCP are defined in other FC-4 documents.

6.2.1 PRLI with IPA required by originator and responder

Operation of the PRLI depends on the originator’s and responder’s requirements for process associators. If IPAs are required by both the originator and responder, then each service parameter page in the PRLI request payload shall identify an originator PA / responder PA image pair. The requirements specified in the page shall be those that correspond to the combined understandings expected to be agreed to by the responder. The capabilities shall be those of the originator only.

In the accept (ACC), the responder shall return exactly one page for each page provided in the PRLI request payload. Each page in the ACC shall identify one of the requested image pairs and shall provide the requirements that are agreed to by the responder. The capabilities presented shall be those for the responder only. Error responses are possible on a page by page basis to indicate that the responder image is unavailable or cannot meet the requested requirements.

Use of this mechanism requires that the originator have precise and detailed knowledge of the requirements and capabilities of each image in the responder. That information may be obtained by mechanisms outside the scope of the FCP or may be obtained by performing a PRLI requesting informative communication.

Service pages are marked as information requests by using request service parameter pages that do not have the originator PA or the responder PA indicated as valid. In that case, the responder returns pages describing the capabilities and requirements of each of its images.

6.2.2 PRLI with IPA required by originator and supported by responder

If IPAs are required by the originator and supported by the responder, then the service parameter pages in the PRLI request shall identify the originator PA and specify the responder PA as invalid. The requirements shall be those expected to be agreed to by the responder and the capabilities shall be those of the originator only.

In the ACC, the service parameter pages shall be returned using the same originator PA and invalid responder PA indication. The requirements shall be set to those that are agreed to by the responder. The capabilities shall be those of the responder for the specified type of operation.

The pages identify exclusive communications between the originator's multiple images (each identified by a process associator) and the single image provided by the responder.

6.2.3 PRLI with IPA supported by originator and required by responder

If IPAs are supported by the originator and required by the responder, then the service parameter pages in the PRLI request shall indicate that both the originator and responder PAs are invalid. The request pages shall specify the requirements expected to be agreed to by the responder. The capabilities shall be those of the originator only.

In the ACC, the service parameter pages shall be returned for each responder PA with the originator PA specified as invalid. The requirements shall be those committed to by the responder. The capabilities shall be those of the responder for the specified type of operation.

The pages identify those exclusive communications established between the single image of the originator and the multiple images (each identified by a process associator) of the responder.

6.2.4 PRLI with IPA not supported by originator or responder

If IPAs are not supported by either the originator or responder, then the service parameter pages in the PRLI request shall indicate that both the originator and responder PAs are invalid. The request pages shall specify the requirements expected to be agreed to by the responder. The capabilities shall be those of the originator only.

In the ACC, the service parameter pages shall be returned with the originator and responder PAs specified as invalid. The requirements shall be those committed to by the responder. The capabilities shall be those of the responder for the specified type of operation.

The pages identify those exclusive communications established between the single image of the originator and the single image of the responder.

6.2.5 New or repeated PRLI

A PRLI affects only those PAs or image pairs specifically referenced by the pages. A new PRLI page to an already established image pair replaces the previous parameters with new PRLI parameters. New PRLI pages can be mixed in any combination with pages that address established image pairs. If the change in parameters affects any outstanding FCP exchanges, those exchanges shall be terminated using a recovery abort operation. A recovery qualifier may be established after the recovery abort, temporarily restricting the choice of OX_ID values. Only actions for image pairs that are being referenced by the PRLI are affected.

Immediately after the execution of a PRLI, both members of the image pair shall have the same state as they would have after a hard reset or a power on with respect to each other. No tasks, reservations or status except those previously established between other image pairs sharing one of the process associators shall be present in either SCSI device. The MODE SELECT parameters will assume their default or saved states for that image pair. Tasks, reservations, status, and MODE SELECT parameters for other initiators are not affected. A Unit Attention condition (Sense Key = 6) with an Additional Sense Code of Reset Occurred (ASC = 29, ASCQ = 00) shall be presented upon the first attempt to communicate between the image pairs using FCP when a new PRLI has been performed. A target port shall not generate a unit attention condition for initiators which are already logged in. Initiators shall indicate to their host systems that an image pair has been reset and that any outstanding operations for the image pair have been reset.

Devices may have default PRLI information provided at the time the device is installed in the configuration. Such devices do not require the execution of a PRLI to perform normal FCP operations. If a device has no such default parameters and if no PRLI has been performed since power on or the last PRLO, any FCP IUs transmitted to that device shall be discarded. If default PRLI information is complete enough so that N_Port login (PLOGI) is sufficient to perform an implicit PRLI, then PLOGI shall establish the same reset state and Unit Attention condition that would normally be established by PRLI.

6.2.6 FCP service parameter page for PRLI request

The FCP service parameter page for the process login request is shown in table 10.

Table 8 - FCP service parameter page, PRLI request

FCP service parameter	Word	Bit
hexadecimal ‘08’, SCSI FCP	0	31–24
reserved for type code extension	0	23–16
ORIGINATOR PROCESS ASSOCIATOR VALID	0	15
RESPONDER PROCESS ASSOCIATOR VALID	0	14
ESTABLISH IMAGE PAIR	0	13
reserved	0	12–0
Originator process associator	1	31–0
Responder process associator	2	31–0
reserved	3	31–7
DATA OVERLAY ALLOWED	3	6
INITIATOR FUNCTION	3	5
TARGET FUNCTION	3	4
COMMAND/DATA MIXED ALLOWED	3	3
DATA/RESPONSE MIXED ALLOWED	3	2
READ XFER_RDY DISABLED	3	1
WRITE XFER_RDY DISABLED	3	0

6.2.6.1 Word 0, Bits 31–24: FCP specific code

The value of hexadecimal ‘08’ in this byte indicates that this service parameter page is defined for FCP.

6.2.6.2 Word 0, Bit 15: ORIGINATOR PROCESS ASSOCIATOR VALID

If this bit is set to *0*, the originator process associator value in word 1 is not meaningful.

If set to *1*, the originator process associator value in word 1 is meaningful.

6.2.6.3 Word 0, Bit 14: RESPONDER PROCESS ASSOCIATOR VALID

If this bit is set to *0*, the responder process associator value in word 2 is not meaningful.

If set to *1*, the originator process associator value in word 2 is meaningful.

6.2.6.4 Word 0, Bit 13: ESTABLISH IMAGE PAIR

If this bit is set to *0*, the PRLI only exchanges service parameters.

If set to *1*, the PRLI exchanges service parameters and establishes an image pair.

6.2.6.5 Word 1: originator process associator

This field is the originator process associator as defined by X3.230-199X and annex A.

6.2.6.6 Word 2: responder process associator

This field is the responder process associator as defined by X3.230-199X and annex A.

6.2.6.7 Word 3, Bit 6: DATA OVERLAY ALLOWED

When this bit is set to *1*, the process defined by the page is indicating that its initiator function has the capability of supporting data overlay. When the bit is set to *0*, the initiator function does not have the capability of performing data overlay. The bit shall be *0* for devices having only target function. If the initiator function supports data overlay, then a target may optionally transfer FCP_DATA IUs that are overlaid. DATA OVERLAY ALLOWED is a PRLI capability that is only defined for the initiator function.

6.2.6.8 Word 3, Bit 5: INITIATOR FUNCTION

When this bit is set to *1*, the process defined by the page is indicating that it operates as a SCSI initiator. When the bit is set to *0*, the process does not operate as a SCSI initiator. INITIATOR FUNCTION is a PRLI capability

6.2.6.9 Word 3, Bit 4: TARGET FUNCTION

When this bit is set to *1*, the process defined by the page is indicating that it operates as a SCSI target. When the bit is set to *0*, the process does not operate as a SCSI target. TARGET FUNCTION is a PRLI capability. Both bits 4 and 5 may be set. If neither bit is set, the service parameters for the page are assumed to be invalid. An originator receiving such an invalid page shall not perform FCP operations with the responder unless implicit login parameters have been established. A responder receiving such an invalid page shall notify the originator with a PRLI ACCEPT RESPONSE CODE of *1000* (Invalid service parameters for page) indication.

6.2.6.10 Word 3, Bit 3: COMMAND/DATA MIXED ALLOWED

When this bit is set to *1*, FCP_CMND and FCP_DATA may be combined in one IU. When this bit is set to *0*, FCP_CMND and FCP_DATA shall be in separate IUs. If both the originator and responder allow the FCP_CMND and FCP_DATA to be combined in one IU, then FCP I/O Operations performing a write operation (an FCP operation transferring data from the initiator to the target) shall use the FCP_CMND and FCP_DATA combined in one IU. If either the originator or the responder does not allow the FCP_CMND and FCP_DATA to be combined, then all FCP I/O Operations between them shall transmit the FCP_CMND and FCP_DATA in separate IUs. If the PRLI FCP service parameters allow the FCP_CMND and FCP_DATA to be combined in one IU, but the corresponding FCP_Port login procedure (PLOGI) does not, the service parameters are invalid. A responder receiving such an invalid page shall notify the originator with a PRLI ACCEPT RESPONSE CODE of *1000* (Invalid service parameters for page) indication for that page.

The valid and invalid combinations of bits 3 and 0 of Word 3 are shown in table 9. A responder receiving such an invalid combination of bits shall notify the originator with a PRLI ACCEPT RESPONSE CODE of *1000* (Invalid service parameters for page) indication for that page.

Table 9 - Valid and invalid combinations of bits 3 and 0 of word 3

CMND/DATA MIXED ALLOWED	WRITE XFER_RDY DISABLED	Valid ?
0	0	Yes
0	1	Yes
1	0	No
1	1	Yes

6.2.6.11 Word 3, Bit 2: DATA/RESPONSE MIXED ALLOWED

When this bit is set to *1*, FCP_DATA and FCP_RSP may be combined in one IU. When this bit is set to *0*, FCP_DATA and FCP_RSP shall be in separate IUs. If both the originator and responder allow the FCP_DATA and FCP_RSP to be combined in one IU, then FCP I/O Operations performing a read operation (an FCP operation transferring data from the target to the initiator) shall be allowed to transmit the last FCP_DATA and FCP_RSP combined in one IU. If either the originator or the responder does not allow the FCP_DATA and FCP_RSP to be combined, then all FCP I/O Operations between them shall transmit the last FCP_DATA and the FCP_RSP in separate IUs. If the PRLI FCP service parameters allow the FCP_DATA and FCP_RSP to be combined in one IU, but the corresponding FCP_Port login procedure does not, the service parameters are invalid. A responder receiving such an invalid page shall notify the originator with a PRLI ACCEPT RESPONSE CODE of *1000* (Invalid service parameters for page) indication.

6.2.6.12 Word 3, Bit 1: READ XFER_RDY DISABLED

When this bit is set to *0*, FCP_XFER_RDY IUs shall be used for SCSI read operations. When this bit is set to *1*, FCP_XFER_RDY IUs may be not used for SCSI read operations. If both the originator and responder choose to disable read FCP_XFER_RDY IUs, then all FCP I/O Operations performing SCSI reads between the FCP_Ports shall operate without using those IUs. If either the originator or the responder requires the use of FCP_XFER_RDY IUs during reads, then the exchange responder shall transmit an FCP_XFER_RDY IU before each FCP_DATA IU is transmitted to the exchange originator.

6.2.6.13 Word 3, Bit 0: WRITE XFER_RDY DISABLED

When this bit is set to *0*, FCP_XFER_RDY IUs shall be used for SCSI write operations. When this bit is set to *1*, FCP_XFER_RDY IUs may be not used before the first FCP_DATA IU to be transferred in the write operation. If both the originator and responder choose to disable write FCP_XFER_RDY IUs, then all FCP I/O Operations performing SCSI writes between the FCP_Ports shall operate without using the FCP_XFER_RDY IU before the first FCP_DATA IU. The FCP_XFER_RDY IU shall be transmitted to request each additional FCP_DATA IU, if any, after the first one. If either the originator or the responder requires the use of FCP_XFER_RDY IUs during writes, then the exchange responder shall transmit an FCP_XFER_RDY IU requesting each FCP_DATA IU, including the first, from the exchange originator.

6.2.7 Service parameters for PRLI accept

The FCP service parameter page for the process login accept is shown in table 10.

Table 10 - FCP service parameter page, PRLI accept

FCP service parameter	Word	Bit
hexadecimal ‘08’, SCSI FCP	0	31–24
reserved for Type Code Extension	0	23–16
ORIGINATOR PROCESS ASSOCIATOR VALID	0	15
RESPONDER PROCESS ASSOCIATOR VALID	0	14
IMAGE PAIR ESTABLISHED	0	13
reserved	0	12
ACCEPT RESPONSE CODE	0	11–8
reserved	0	7–0
Originator process associator	1	31–0
Responder process associator	2	31–0
reserved	3	31–6
INITIATOR FUNCTION	3	5
TARGET FUNCTION	3	4
COMMAND/DATA MIXED ALLOWED	3	3
DATA/RESPONSE MIXED ALLOWED	3	2
READ XFER_RDY DISABLED	3	1
WRITE XFER_RDY DISABLED	3	0

With the following exceptions, the service parameter definitions are identical for the PRLI request and accept pages.

6.2.7.1 Word 0, Bit 13: IMAGE PAIR ESTABLISHED

IMAGE PAIR ESTABLISHED is valid only if bit 13 was set to 1 on the corresponding Service Parameter page of the PRLI request.

If this bit is set to *0*, the image pair was not established. The **ACCEPT RESPONSE CODE** has additional information.

If set to *1*, the image pair was established.

6.2.7.2 PRLI ACCEPT RESPONSE CODE

This 4-bit value is defined in annex A. The value indicates whether the image pair was successfully created. If the image pair could not be created, the response code indicates why the request failed or was rejected.

6.3 Process logout (PRLO)

The process logout (PRLO) request is transmitted from an originator FCP_Port to a responder FCP_Port to indicate to the responder that those process image pairs specified in the service parameter pages of the PRLO are being discontinued by the originator. All tasks, reservations, mode page parameters, and status for image pairs removed by the PRLO operation are set to the state they would have after a SCSI hard reset or power on reset. Only the specified image pairs are logged out. Open exchanges for logged out image pairs shall be terminated by a recovery abort operation. (See 7.1.2.5) Tasks, reservations, mode page parameters, and status for image pairs other than those that are logged out are not affected.

A PRLO page identifies an image pair where only the originator or only the responder requires IPA by marking only the originator PA or only the responder PA as valid. The image pair that is specified by the single PA and the opposite N_Port is logged out. No further FCP communication is possible between these two images. Those sequences and exchanges outstanding between those image pairs shall be terminated by a recovery abort operation.

A PRLO page identifies an image pair where neither the originator or responder supports IPA by marking the originator PA and responder PA as invalid. No further FCP communication is possible between those two N_Ports. Those sequences and exchanges outstanding between those image pairs shall be terminated by a recovery abort operation.

The PRLO accept (ACC) is returned to the originator FCP_Port to indicate that the responder FCP_Port recognizes that the image pairs are being discontinued. The ACC shall present a response service parameter page for every request service parameter page. It is not an error to perform a PRLO for an image pair that is not known to the responder.

A link service reject (LS_RJT) indicates that the PRLO request is invalid and not accepted.

Unless implicit login parameters exist that allow FCP functions and establish default parameters, the PRLO shall set the image pair to a state that does not allow any FCP initiator or target functions to be initiated or performed. Attempts to communicate with an image that has not been logged in or has been logged out shall be acknowledged in the normal manner. The responder shall then perform a PRLO operation for the image pair to indicate that the PA has been logged out.

The PRLO common service parameters and ACCEPT RESPONSE CODES are defined in annex A.

6.3.1 FCP service parameter page for PRLO request

The FCP service parameter page for the process logout request is shown in table 10.

Table 11 - FCP service parameter page, PRLO request

FCP service parameter	Word	Bit
reserved	0	31–24
reserved	0	23–16
ORIGINATOR PROCESS ASSOCIATOR VALID	0	15
RESPONDER PROCESS ASSOCIATOR VALID	0	14
reserved	0	13–0
Originator process associator	1	31–0
Responder process associator	2	31–0
reserved	3	31–0

All the fields in the parameter page have the same definitions as the field of the same name in the PRLI request page.

6.3.2 FCP service parameter page for PRLO accept

The FCP service parameter accept page for the process logout is shown in table 10.

Table 12 - FCP service parameter page, PRLO accept

FCP service parameter	Word	Bit
reserved	0	31–24
reserved	0	23–16
ORIGINATOR PROCESS ASSOCIATOR VALID	0	15
RESPONDER PROCESS ASSOCIATOR VALID	0	14
reserved	0	13–0
ACCEPT RESPONSE CODE	0	11–8
reserved	0	7–0
Originator process associator	1	31–0
Responder process associator	2	31–0
reserved	3	31–0

All the fields in the parameter page have the same definitions as the field of the same name in the PRLI accept page except as indicated below.

6.3.2.1 PRLO ACCEPT RESPONSE CODE

This 4-bit value is defined in annex A. The value indicates whether the image pair for which the logout was successful. The failure indications are presently limited to an indication that the image pair did not exist and an indication that the destination N_Port could not process a request with multiple service parameter pages.

7 FCP information unit (IU) formats

This clause describes the contents of the IUs transferred during FCP I/O Operations.

7.1 FCP_CMND IU

The FCP_CMND IU carries either a SCSI Command to be executed or a task management request to be performed. The FCP_CMND IU shall contain the values and control fields defined in table 13 in its payload.

Table 13 - FCP_CMND payload

Field Name	Description	Size
FCP_LUN	Logical Unit Number	8 bytes
FCP_CNTL	Control Field	4 bytes
FCP_CDB	SCSI command descriptor block	16 bytes
FCP_DL	Data Length	4 bytes

If XFER_RDY IUs are disabled and Command/Data Mixed is allowed, the FCP_CMND and the first or only FCP_DATA burst shall be combined in a single IU during SCSI write operations. In all other cases, the FCP_CMND IU shall be transmitted in a separate IU that transfers initiative to the exchange responder.

7.1.1 FCP_LUN

The FCP logical unit number is the address of the desired logical unit in the attached subsystem. The FCP_LUN field is specified by X3.230-199X for all IUs of Category 6.

Each SCSI FCP target shall accept an INQUIRY command to the logical unit addressed by the FCP_LUN zero (0000 0000 0000 0000 hexadecimal). Using the SCSI INQUIRY information, the initiator can determine the SCSI device type, manufacturer, and model of the logical unit. If the logical unit at LUN 0 specifies a device model that has a defined addressing structure, the initiator can use that information to determine what other logical units are implemented in the target. The structure of the FCP_LUN field is not specified by the FCP. The structure is specified by the SCSI device model and may be vendor unique for device models that do not have a defined address structure. An example of a four-layer hierarchical address is given in annex C.

If the FCP_LUN address locates a valid logical unit, the command shall be executed according to standard SCSI behavior. Behavior may include successful execution of the command, presentation of errors associated with the command, or rejection of the command. If the addressed logical unit does not exist, the responses shall follow the SCSI-3 rules for selection of invalid logical units.

7.1.2 FCP_CNTL

The FCP_CNTL field contains a number of control flags and control bits arranged as shown in table 14.

Table 14 - FCP_CNTL field format

Bit Definition	Bit
Byte 0 (Most Significant), Reserved	
reserved	7–0
Byte 1, Task Codes	
reserved	7–3
TASK ATTRIBUTE	2–0
Byte 2, Task management flags	
TERMINATE TASK	7
CLEAR ACA	6
TARGET RESET	5
reserved	4,3
CLEAR TASK SET	2
ABORT TASK SET	1
reserved	0
Byte 3, Execution management codes	
reserved	7–2
READ DATA	1
WRITE DATA	0

7.1.2.1 Task Codes, Byte 1

TASK ATTRIBUTE values are shown in table 15.

Table 15 - TASK ATTRIBUTE values

Value, bits 2–0	TASK ATTRIBUTE
000 b	SIMPLE_Q
001 b	HEAD_OF_Q
010 b	ORDERED_Q
100 b	ACA_Q
101 b	UNTAGGED
others	reserved

SIMPLE_Q requests that the task be managed according to the rules for a SIMPLE task attribute.

HEAD_OF_Q requests that the task be managed according to the rules for a HEAD OF QUEUE task attribute.

ORDERED_Q requests that the task be managed according to the rules for an ORDERED task attribute. With a class 2 fabric, special care must be exercised to guarantee successful ordering. Sequential delivery must be requested at login to ensure correct ordering among tasks. FCP_CMND IUs must be acknowledged before new FCP_CMND IUs are issued to avoid inadvertent reordering of commands during retries of F_BSY. Ordering can also be accomplished by waiting for the completion of those commands requiring ordering before transmitting the FCP_CMND for the next FCP I/O operation.

ACA_Q requests that the task be managed according to the rules for an automatic contingent allegiance (ACA) task attribute.

UNTAGGED requests that the task be managed according to the rules for an untagged task. Only one untagged task can exist for each logical unit identifier / initiator pair. Requesting a second untagged command for the same logical unit identifier / initiator pair shall be treated as an overlapped command. [X3.270-199X] An untagged task is scheduled according to the rules for a SIMPLE task attribute.

7.1.2.2 Task management flags, Byte 2

Except for TERMINATE TASK, a Task management function shall be transmitted by the initiator (Exchange Originator) using a new Exchange. If any task management flag is set to *1*, the FCP_CDB, FCP_DL, and CDB related FCP_CNTL flags (task codes and execution management codes) are not valid and are ignored. No more than one task management flag shall be set to *1* in any FCP_CMND IU.

TERMINATE TASK requests that the specified task be terminated without corrupting the medium. This bit is optional for FCP devices. The TERMINATE TASK operation shall be sent by the initiator using the FQXID for the task to be terminated. If necessary, the initiator (exchange originator) must first perform a request sequence initiative link service request to gain control of the exchange. The TERMINATE TASK is then sent using an FCP_CMND IU.

CLEAR ACA, when set to *1*, causes the ACA condition to be cleared. When the task manager clears the auto contingent allegiance condition, any task within that task set may be completed subject to the rules for task management specified by X3.270-199X.

When set to *0*, the ACA condition remains unchanged.

The use of the ACA bit in the CDB control field and the implementation of ACA is described in X3.270-199X.

If the ACA bit in the CDB control field is set to *0*, the automatic sense operation performed by the presentation of the FCP_RSP IU shall clear the ACA condition. Depending on the MODE SELECT parameters that have been established, additional FCP I/O operations may have to be aborted by the recovery abort

The CLEAR ACA is transmitted by the initiator (exchange originator) using a new exchange.

TARGET RESET, when set to *1*, performs a reset to the SCSI device as defined in X3.270-199X. TARGET RESET resets all tasks for all initiators and resets all internal states of the target to their initial power on and default values as established by PRLI. A unit attention condition is created for all initiators. The FCP login state of the affected image pairs is not changed by the TARGET RESET.

The TARGET RESET is transmitted by the initiator (exchange originator) using a new exchange. The initiator and target clear all resources that can be cleared unambiguously. Any open exchanges that are in an ambiguous state shall be terminated by whichever port detects the ambiguous state using a recovery abort. The ports may issue additional recovery abort operations if they are unable to determine in a simple manner whether the state of an FCP I/O operation is ambiguous.

For a target or initiator FCP_Port, an exchange is in an ambiguous state if the FCP_Port has sequence initiative and there exists an unacknowledged frame for the sequence or if the FCP_Port has transferred sequence initiative but the transfer of the initiative has not been confirmed. For a target FCP_Port, an exchange is also in an ambiguous state if the exchange exists between the target FCP_Port and an initiator other than the initiator FCP_Port that performed the TARGET RESET.

CLEAR TASK SET causes all tasks from all initiators in the specified task set to be aborted as defined in X3.270-199X. A unit attention condition is created for all initiators other than the initiator that sent the CLEAR TASK SET that had tasks in the task set.

The CLEAR TASK SET is transmitted by the initiator (exchange originator) using a new exchange. The initiator and target clear any resources that can be cleared unambiguously. Any open exchanges that are in an ambiguous state shall be terminated by whichever port detects the ambiguous state using a recovery abort. The ports may issue additional recovery abort operations if they are unable to determine in a simple manner whether the state of an FCP I/O operation is ambiguous.

For a target or initiator FCP_Port, an exchange is in an ambiguous state if the FCP_Port has sequence initiative and there exists an unacknowledged frame for the sequence or if the FCP_Port has transferred sequence initiative but the transfer of the initiative has not been confirmed. For a target FCP_Port, an exchange is also in an ambiguous state if the exchange exists between the target FCP_Port and an initiator other than the initiator FCP_Port that performed the CLEAR TASK SET.

ABORT TASK SET causes all tasks in the task set from the initiator requesting the ABORT TASK SET to be aborted as defined in X3.270-199X. The bit is mandatory in FCP.

The ABORT TASK SET is transmitted by the initiator (exchange originator) using a new exchange. The initiator and target clear any resources that can be cleared unambiguously. Any open exchanges that are in an ambiguous state shall be terminated by whichever port detects the ambiguous state using a recovery abort. The ports may issue additional recovery abort operations if they are unable to determine in a simple manner whether the state of an FCP I/O operation is ambiguous.

For a target or initiator FCP_Port, an exchange is in an ambiguous state if the FCP_Port has sequence initiative and there exists an unacknowledged frame for the sequence or if the FCP_Port has transferred sequence initiative but the transfer of the initiative has not been confirmed.

> NOTE - X3T10 plans to consider recommendations from industry groups and other contributors resulting from early implementations and evaluation of FCP (e.g. a definition of the Abort Task Set, Clear Task Set, and Target Reset functions, in FCP, which may eliminate associated recovery abort functions generated by a target) in a future version of FCP.

7.1.2.3 Other task management functions

ABORT TASK causes the target to abort the specified task, if the task exists. The action is defined in X3.270-199X. The ABORT TASK is performed by the initiator (exchange originator) using the FQXID of the exchange to be aborted. Any exchange shall be terminated by the initiator using the recovery abort. To be compliant with X3.230-199X, the ABORT TASK may not immediately release all exchange resources, since a recovery qualifier may be established to allow for the management of information that may already have been delivered to the fabric.

7.1.2.4 Execution management codes, Byte 3

READ DATA, when set to *1*, specifies that the initiator expects FCP_DATA IUs for the task to be in the direction opposite to the direction of the FCP_CMND IU. This is a SCSI read-type operation.

WRITE DATA, when set to *1*, specifies that the initiator expects FCP_DATA IUs for the task to be in the same direction as the FCP_CMND IU. This is a SCSI write operation. If both READ DATA and WRITE DATA are set to *0*, there shall be no FCP_DATA IUs and FCP_DL shall be *0*. The initiator shall not set both the READ DATA and the WRITE DATA bits to *1*.

7.1.2.5 Recovery abort

The recovery abort is an FC-PH protocol that recovers FCP_Port resources associated with an exchange that is being terminated. The following protocol has been selected for simplicity, completeness, and robustness.

> The FCP_Port wanting to terminate the exchange generates an ABTS sequence. The ABTS sequence is generated using the OX_ID and RX_ID of the exchange to be aborted. The FC-PH allows ABTS to be generated by an FCP_Port regardless of whether or not it has sequence initiative.
>
> If the ABTS is accepted, the BA_ACC frame shall contain the L_S bit set in the F_CTL field, indicating that the exchange ends at the end of the sequence. A recovery qualifier is established if necessary to discard any pending frames for the exchange and to prevent the reuse of the OX_ID and RX_ID for at least R_A_TOV. The BA_ACC shall request that the recovery qualifier cause all frames for all sequences of the exchange to be discarded by setting SEQ_CNT_LO to 0 and SEQ_CNT_HI to “FFFF” hexadecimal. If the recovery qualifier was established, after R_A_TOV, the transmitter of the ABTS shall send a reinstate recovery qualifier (RRQ) link service request to again allow use of the OX_ID and RX_ID.
>
> A target shall always accept an ABTS using the unassigned RX_ID value of “FFFF” hexadecimal and establish a recovery qualifier with a specified RX_ID.
>
> An ABTS with an assigned RX_ID value may be rejected with a reason code of “logical error/invalid OX_ID/RX_ID combination” if the exchange has already been cleared. If the ABTS is rejected, the exchange has already been successfully terminated, all exchange resources recovered and no further action is required. The BA_RJT frame contains the L_S bit set in the F_CTL field, indicating that the exchange opened by the ABTS ends at the end of the sequence.

In addition to recovering exchange resources that may have been left unavailable while executing task management functions, recovery abort can be used to recover exchange resources left in an undefined state by any of the task abort events defined in X3.270-199X or by any similar events.

7.1.3 FCP_CDB

The FCP_CDB field contains the actual CDB to be interpreted by the addressed logical unit. The maximum CDB length is 16 bytes. The FCP_CDB is not valid and is ignored if any task management flag is set to *1*.

The command byte of the CDB shall be located at the first byte transmitted of the FCP_CDB field. The CDB's first byte defines the length and basic format of the remainder of the CDB as described in X3.270-199X. Bytes beyond the last byte of the CDB are not defined by FCP, shall be ignored by the target, and may have any value.

7.1.4 FCP_DL

The FCP_DL field contains a count of the greatest number of data bytes expected to be transferred to or from the application client data buffer by the SCSI CDB. The parameter is the command byte count defined by X3.270-199X. An FCP_DL value of *0* indicates that no data transfer is expected regardless of the state of the READ DATA and WRITE DATA bits and that no FCP_XFER_RDY or FCP_DATA IUs shall be transferred.

7.2 FCP_XFER_RDY IU

The FCP_XFER_RDY IU indicates that the target is prepared to perform part or all of the data transfer for a command. The FCP_XFER_RDY IU contains those parameters of the SCSI-3 data delivery service required by the initiator.

FCP_XFER_RDY IUs shall be transmitted preceding each read-type and/or write-type FCP_DATA IU when the corresponding XFER_RDY_Disabled bit is set to *0* by PRLI. If the target and initiator have negotiated WRITE XFER_RDY DISABLED, FCP_XFER_RDY IUs shall be transmitted to request each write-type FCP_DATA IU after the first of one of a SCSI command.

During write operations, the FCP_XFER_RDY IU indicates the amount of data that the target expects to be transferred from the initiator. Since the target has planned buffering and caching resources based on that amount of data, the initiator shall provide precisely that amount of data. The initiator shall be ready to transmit the entire FCP_DL bytes of data.

During read operations, the FCP_XFER_RDY IU indicates the amount of data that the target shall transfer to the initiator. The initiator shall have available enough data buffer space to contain the entire data transfer with no more warning than the receipt of the FCP_XFER_RDY IU. The FCP_XFER_RDY IU provides a warning to the initiator that it is about to receive a burst of data.

The first 8 bytes of the FCP_XFER_RDY payload are defined by X3.230-199X for all IUs of category 5. The format of the FCP_XFER_RDY payload is shown in table 16.

Table 16 - FCP_XFER_RDY payload

Field Name	Description	Size
DATA_RO	Relative offset of first byte of FCP_DATA IU that follows	4 bytes
BURST_LEN	Length of FCP_DATA IU that follows	4 bytes
reserved		4 bytes

7.2.1 DATA_RO

The DATA_RO field indicates the contents of the RLTV_OFF field for the first data byte of the next FCP_DATA IU. This may be used by the responder to deliver data out of order on reads and to request data out of order on writes. This is the same as the SCSI-3 application client buffer offset.

The proposed standard ANSI X3T10/995D specifies a mechanism for turning off the capability of modifying data pointers to perform out of order data delivery using a bit defined in the MODE SENSE/SELECT disconnect-reconnect page. X3.269 defines this mechanism in the following manner. If bit 7 of byte 12 (the ENABLE MODIFY DATA POINTERS or EMDP bit) is set to a *1* value in the disconnect-reconnect page, the target is allowed to generate consecutive DATA_RO values that are not continuously increasing within a single FCP I/O Operation. If the EMDP bit is set to a *0* value, then the target shall generate continuously increasing DATA_RO values within a single FCP I/O Operation.

7.2.2 BURST_LEN

For data transfers from the SCSI initiator to the target, the BURST_LEN field indicates the amount of buffer space prepared for the next FCP_DATA IU and requests the transfer of an IU of that exact length. This value is the same as the SCSI data delivery request byte count.

For data transfers from the SCSI target to the initiator, the BURST_LEN field indicates the exact size of the FCP_DATA IU that shall follow the FCP_XFER_RDY IU. The value in this field shall not exceed the maximum burst length defined by the disconnect/reconnect page of MODE SELECT/MODE SENSE. (See 7.3)

A BURST_LEN of *0* is not valid.

7.3 FCP_DATA IU

The data associated with a particular FCP I/O Operation is identified by the FQXID.

SCSI data transfers may be performed by one or more data delivery requests, each one performing a transfer no longer than the maximum burst length defined by the parameters of the disconnect/reconnect page. The disconnect/reconnect page is examined and set by the MODE SENSE and MODE SELECT commands.

If more than one FCP_DATA IU is used to transfer the data, the SEQ_ID and the RLTV_OFF are used to ensure that the SCSI data is reassembled in the proper order. If an FCP_XFER_RDY IU is used to describe a data transfer and the subsequent FCP_DATA IU has a lowest RLTV_OFF that differs from the DATA_RO of the FCP_XFER_RDY, the target shall post the error code indicating an RO mismatch in the FCP_RSP_INFO field of the FCP_RSP.

If required by the PRLI FCP service parameters, each inbound and/or outbound FCP_DATA IU shall be preceded by an FCP_XFER_RDY IU containing a standard data descriptor payload that indicates the exact location and length of the data delivery. If the PRLI FCP service parameters specify READ XFER_RDY DISABLED and/or WRITE XFER_RDY DISABLED, the corresponding FCP_DATA IUs are transmitted without a preceding FCP_XFER_RDY IU.

During any data transfer, the initiator shall always have available a buffer of length FCP_DL. The buffer contains data to be transferred to the target if the operation is an outbound or write operation. The buffer receives the data if the operation is an inbound or read operation. The target shall never request or deliver data outside the buffer length defined by FCP_DL. If the command requested that data beyond FCP_DL be transferred, the FCP_STATUS field shall contain the FCP_RESID_OVER bit. The command is completed normally except for presentation of the overrun condition. See 7.4.1.

During a write operation that is not using FCP_XFER_RDY IUs, the initiator indicates that it has transferred all the required data by transferring initiative to the target. The initiator shall not transfer data outside the buffer length defined by FCP_DL. If the write operation requires a total amount of data less than the amount of data provided by the initiator, the target shall discard the excess bytes and indicate that an overrun has occurred by setting the FCP_RESID_OVER bit in the FCP_STATUS field. The command is completed normally except for presentation of the overrun condition.

If the PRLI service parameter DATA OVERLAY ALLOWED for the initiator is *1*, the target may request that data be overlaid. If the PRLI service parameter DATA OVERLAY ALLOWED for the initiator is *0*, the target shall not request that data be overlaid. If data overlay is not allowed and the target attempts to overlay data, the initiator may not be able to guarantee data integrity and may indicate service delivery failure.

The target may request data bursts in any order. By the time data transfer has been terminated, all data between the offset of *0* and the highest offset shall have been transferred. The target shall not request that sets of data in the middle of a transfer not be transferred. If error conditions occur that prevent the transfer of a set of data in the middle of a data transfer, the FCP_SNS_INFO shall indicate that only data from the offset of *0* to the highest offset before the untransferred data space has been transferred. Even if data of a higher offset was successfully transferred, it shall not be considered valid.

X3.230-199X specifies the mechanisms by which an IU shall be transferred. The mechanisms vary with which Class of Service is being used and what service parameters are in effect.

7.4 FCP_RSP IU

The content of the FCP_RSP IU is indicated in table 17.

Table 17 - FCP_RSP IU

Field Name	Description	Size
reserved		4 bytes
reserved		4 bytes
FCP_STATUS	Field validity and SCSI status	4 bytes
FCP_RESID	Residual Count	4 bytes
FCP_SNS_LEN	Length of FCP_SNS_INFO field	4 bytes
FCP_RSP_LEN	Length of FCP_RSP_INFO field	4 bytes
FCP_RSP_INFO	FCP Response Information	m bytes
FCP_SNS_INFO	SCSI Sense Information	n bytes

7.4.1 FCP_STATUS

The FCP_STATUS field is normally *0* upon successful completion of a FCP I/O Operation. If command linking is being performed, an FCP_RSP IU is provided for each command executed. A value of *0* means no error, and in addition indicates that no other information is present in the FCP_RSP. For linked commands, INTERMEDIATE status or INTERMEDIATE - CONDITION MET status indicates successful completion of a command with no other information valid if all other fields are *0*.

The format of the FCP_STATUS field is shown in table 18.

.

Table 18 - FCP_STATUS field format

Bit Definition	Bit
Byte 0 (Most Significant)	
reserved	7–0
Byte 1	
reserved	7–0
Byte 2	
reserved	7–4
FCP_RESID_UNDER	3
FCP_RESID_OVER	2
FCP_SNS_LEN_VALID	1
FCP_RSP_LEN_VALID	0
Byte 3	
SCSI status byte	7–0

Bytes 0 (MSB) and 1 are reserved and set to *0*.

Byte 2 contains validity and status indicators for the subsequent fields.

> FCP_RESID_UNDER indicates that the FCP_RESID field is valid and contains the count of bytes that were expected to be transferred, but were not transferred.
>
> FCP_RESID_OVER indicates that the FCP_RESID field is valid and contains the count of bytes that could not be transferred because the FCP_DL was not sufficient.
>
> FCP_SNS_LEN_VALID indicates that the FCP_SNS_LEN field is valid and contains the count of bytes in the FCP_SNS_INFO field.
>
> FCP_RSP_LEN_VALID indicates that the FCP_RSP_LEN field is valid and contains the count of bytes in the FCP_RSP_INFO field.

Byte 3 (LSB) contains the status byte from the SCSI logical unit. The status byte codes are defined by X3.270-199X.

7.4.2 FCP_RESID

If the FCP_RESID_UNDER or the FCP_RESID_OVER bit is *1*, the FCP_RESID field contains a count of the number of residual data bytes which were not transferred in the FCP_DATA IUs for this SCSI command. Upon successful completion of a FCP I/O Operation, the residual value is normally *0* and the FCP_RESID value is not valid. Devices having indeterminate data lengths may have a nonzero residual byte count after completing valid operations. Targets are not required to verify that the data length implied by the contents of the CDB will create an overrun or underrun before beginning execution of an SCSI command.

If the FCP_RESID_UNDER bit is set, a transfer that did not fill the buffer to the expected displacement FCP_DL was performed and the value of FCP_RESID is a number equal to:

FCP_DL - highest offset of any byte transmitted

A condition of FCP_RESID_UNDER may not be an error for some devices and some commands.

If the FCP_RESID_OVER bit is set, the transfer was truncated because the data transfer required by the SCSI command extended beyond the displacement value of FCP_DL. Those bytes that could be transferred without violating the FCP_DL value may be transferred. The FCP_RESID is a number equal to:

(Transfer length required by command) - FCP_DL

If a condition of FCP_RESID_OVER is detected, the termination state of the FCP I/O operation is not certain. Data may or may not have been transferred and the SCSI status byte may or may not provide correct command completion information.

If the FCP_RESID_UNDER and the FCP_RESID_OVER bits are *0*, the FCP_RESID field is not meaningful and may have any value.

7.4.3 FCP_SNS_LEN

If the FCP_SNS_LEN_VALID bit is *1*, the FCP_SNS_LEN field specifies the number of valid bytes of FCP_SNS_INFO.

If the FCP_SNS_LEN_VALID bit is *0*, the FCP_SNS_LEN field is not valid and no FCP_SNS_INFO is provided.

The FCP_SNS_LEN field is always included in the FCP_RSP IU.

7.4.4 FCP_RSP_LEN

If the FCP_RSP_LEN_VALID bit is *1*, the FCP_RSP_LEN field specifies the number of valid bytes of FCP_RSP_INFO. The number shall be 0, 4, or 8. Other values of length are reserved for future standardization. An FCP_RSP_LEN of *0* specifies that no bytes of response information are being provided.

If the FCP_RSP_LEN_VALID bit is *0*, the FCP_RSP_LEN field is not valid and no FCP_RSP_INFO is provided.

The FCP_RSP_LEN field is always included in the FCP_RSP IU.

7.4.5 FCP_RSP_INFO

The FCP_RSP_INFO field contains information describing only the protocol failures detected during the execution of a FCP I/O Operation. The FCP_RSP_INFO does not contain link error information, since FC-PH provides the mechanisms for presenting such errors. The FCP_RSP_INFO does not contain SCSI logical unit error information, since that is contained in the FCP_SNS_INFO field as described in 7.4.6. The FCP_RSP_INFO field shall contain valid information if the target detects any of the conditions indicated by an FCP RSP_CODE. The format of the FCP_RSP_INFO field is specified in table 19.

Table 19 - FCP_RSP_INFO field format

Bit Definition		Bit
Byte 0 (Most Significant)		
	reserved	7–0
Byte 1		
	reserved	7–0
Byte 2		
	reserved	7–0
Byte 3		
	RSP_CODE	7–0
Bytes 4–7	reserved	7–0

The valid RSP_CODE values are specified in table 20

Table 20 - RSP_CODE definitions

RSP_CODE definition	Value (hexadecimal)
No Failure or Task Management function complete	00
FCP_DATA length different than BURST_LEN	01
FCP_CMND Fields Invalid	02
FCP_DATA RO mismatch with FCP_XFER_RDY DATA_RO	03
Task Management Function Not Supported	04
Task Management Function Failed	05
reserved	06–FF

The task management function may or may not have been performed by the target if RSP_CODE is returned or if no FCP_RSP is returned before the Exchange is aborted. Values 04 and 05 are not valid responses to SCSI commands.

7.4.6 FCP_SNS_INFO

The FCP_SNS_INFO field contains the information specified by X3.131-1994 for presentation by the REQUEST SENSE command. The proper FCP_SNS_INFO shall be presented when the SCSI status byte of CHECK

CONDITION or COMMAND TERMINATED is presented as specified by X3.270-199X. FCP devices should always use autosense.

Annex A
Extended link services

(normative)

This annex contains the specification of FC-PH optional extended link services required for FCP support. These services are generic to systems designed to support multiple FC-4s. Final specification of these functions will be moved to future a Fibre Channel standard, at the time that document is approved as a standard. The usage of extended link services by FCP is specified in the corresponding body of this document.

A.1 Extended link service command codes

When TYPE indicates EXTENDED LINK SERVICE, byte 0 of the first word of the first frame of the payload (LS_Command code) of the request or reply sequence shall contain the encoded value shown in table A.1. Subsequent frames, if any, for a request or reply sequence shall only contain additional payload in the payload field (i.e., the LS_Command code is not repeated In each frame).

Table A.1 - Extended link service command codes

Encoded Value (Bits 31–24)	Description	Abbr.
0010 0000	Process login	PRLI
0010 0001	Process logout	PRLO

Information units associated with the extended link services specified in table A.1 are described in table A.2.

Table A.2 - Extended link service information units

IU Name	Category	Content	F/M/L	SI
PRLI	Unsol control	Process login	F	T
PRLO	Unsol control	Process logout	F	T
Legend: **IU Name** is the information unit name **Category** is the information category **Content** is the payload content **F/M/L** is the **F**irst/**M**iddle/**L**ast sequence of an exchange **SI** is sequence Initiative - **H**eld or **T**ransferred				

A.2 Process login

The process login (PRLI) extended link service request shall be used to establish the operating environment between a group of related processes at the originating N_Port and a group of related processes at the responding N_Port.

Establishing the operating environment may include the establishment of image pairs and the exchange of service parameters. The establishment of image pairs is FC-4 independent and is system structure dependent. The exchange of service parameters is FC-4 dependent, and if required by a particular FC-4, shall be specified in the corresponding FC-4 document.

Process login is long lived. The number of concurrent process logins in effect at an N_Port is a function of the N_Port facilities available. Process login is separate from N_Port login. Process login may be either implicit or explicit.

A.2.1 Implicit process login

Implicit process login is a method of establishing an operating environment by means other than the explicit use of the PRLI exchange. Specific methods of implicit process login are not defined.

A.2.2 Explicit process login

Explicit process login is accomplished by using the PRLI extended link service sequence within a separate exchange to establish an operating environment.

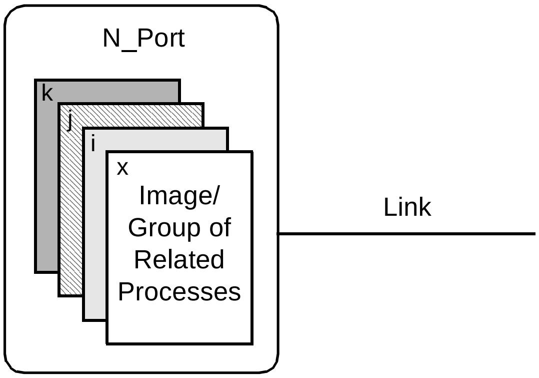

Figure A.1 - Images/groups of related processes

A group of related processes is known as an ‘image’ and is identified by the process associator. The combination of the D_ID, responder PA, S_ID and originator PA identify the image pair. Either a single group or multiple groups of related processes may exist behind an N_Port. A single group of related processes behind an N_Port may be denoted by either an N_Port ID only or N_Port ID and process associator.

PRLI, if required, is performed after N_Port login is successful and prior to other IU transfers. Examples of use of the process login function may include image initialization, image re-configuration, or when the N_Port receives an indication that the image pair no longer exists. PRLI allows each image behind an N_Port to separately manage its resources.

PRLI may be used to establish an operating environment between any of the following combinations of N_Port facilities:

Two N_Ports;

One N_Port and one N_Port image;

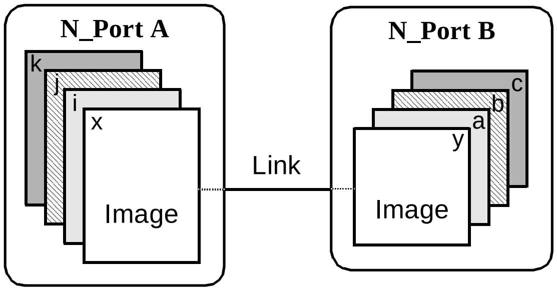

Figure A.2 - Image pairs

Two N_Port images.

Multiple image pairs may be established with a single PRLI request and accept reply sequence set. Failure to establish a particular image pair does not affect existing image pairs or the ability to establish other image pairs.

PRLI may also be used to exchange service parameters without establishing image pairs. However, if an image pair is currently established, a subsequent PRLI request targeted to the same N_Port pair shall identify an image pair in order to modify service parameter settings for that image pair.

A ULP may choose to establish or modify the operating environment for multiple image pairs with a single PRLI request. This can be accomplished by specifying multiple service parameter pages within a single PRLI request. However, a destination N_Port may be unable to execute a PRLI request containing multiple service parameter pages. No protocol to determine the capability of a PRLI responder to execute PRLI requests containing multiple service parameter pages is specified. An accept response containing a nonzero response code is returned in the case that a PRLI responder is unable to execute a PRLI request containing multiple service parameter pages. (See table A.7).

If a PRLI request is received for an established image pair, the existing image pair is unaffected and the PRLI request is processed normally. This allows the exchange of service parameters for an FC-4 not specified when the original image pair was established. PRLO shall be used to remove an established image pair.

It shall be the responsibility of the ULPs to ensure that all active operations over an image pair have been properly terminated prior to issuing a PRLI request that replaces service parameters. If the replacement of service parameters affects any active operations, all active sequences shall be terminated by invoking abort sequence protocol (ABTS). Following the completion of ABTS protocol, all active exchanges shall be terminated by invoking the Abort exchange (ABTX) protocol or the FCP recovery abort. Whether or not the replacement of service parameters affects an active operation shall be specified for each service parameter by the associated FC-4.

The N_Port originating the PRLI request shall not consider the image pair to be established until it has taken the necessary action to establish the image pair, and has received an accept reply sequence indicating that the image pair has been established. The N_Port responding to the PRLI request does not consider the image pair to be established until the necessary action is taken at the N_Port to establish the image pair, and an accept reply sequence is sent.

If a link error is detected when a PRLI request is received, the appropriate response, if any, is made, and the image pair is not established. If the PRLI request cannot be accepted for reasons other than a link-error or link-busy condition, an LS_RJT reply sequence containing the appropriate LS_RJT reason code is sent in response. If an LS_RJT is sent in response to a PRLI request for an image pair that is already established, the existing image pair is unaffected. If an LS_RJT, link-busy, or link-reject response is received to a PRLI request, the PRLI request may be retried until the image pair is established. The number of retries is system dependent. In the case of LS_RJT, whether or not the PRLI is retried depends on the LS_RJT reason code.

In the event that there is an error in the response to establish an image pair, the originating N_Port cannot assume that the requested action has or has not taken place. If no valid response is received by the N_Port to the PRLI request, the N_Port can retry the request. The number of retries is system dependent.

Protocol:

Process login request sequence

Accept reply sequence

Format: FT-1

Addressing: The S_ID field designates the N_Port requesting process login. The D_ID field designates the destination N_Port of the process login.

Payload: The format of the payload is shown in table A.3.

Table A.3 - PRLI payload

Item	Size (Bytes)
x‘20’ = PRLI command code	1
x‘10’ = Page length	1
Payload length	2
Service parameter pages	16 – max

Page length: Byte 1 of word 0 contains an 8-bit unsigned binary integer that specifies the length of each service parameter page. The rightmost two bits shall be zeros.

Payload length: Bytes 2–3 of word 0 contain a 16-bit unsigned binary integer that specifies the length of the PRLI payload. The rightmost two bits shall be zeros. The value specified shall be greater than or equal to 20, and less than or equal to 65,532.

Service parameter page(s): Words 1 through the maximum of the PRLI payload contain one or more service parameter pages. Each service parameter page contains service parameters for a single image pair and is associated with either a single FC-4 type or is common to all FC-4 types for that image pair.

The format of PRLI service parameter pages is described in table A.4.

- **Word 0, Bits 31–24 - TYPE code or common service parameters**

  Identifies the protocol associated with this service parameter page. If byte 0 of the first word of a service parameter page contains the value x‘00’, the service parameter page contains service parameters common to all FC-4 types at that image pair or N_Port pair. If byte 0 of the first word of a service parameter page contains the value other than x‘00’, the service parameter page contains service parameters for the FC-4 type indicated as specified by ANSI X3.230, FC-PH.

- **Word 0, Bits 23–16 - TYPE code extension**

  Reserved for future use.

Table A.4 - PRLI service parameter page format

Item	Word	Bit
Service parameter page	0–3	31–0
TYPE code or common service parameters (1)	0	31–24
TYPE code extension	0	23–16
Originator process associator valid	0	15
Responder process associator valid	0	14
Establish image pair	0	13
Reserved	0	12–0
Originator process associator	1	31–0
Responder process associator	2	31–0
Service parameters	3	31–0
NOTE 1 – If byte 0 of the first word of a service parameter page is set to the value x’00’, the service parameter page is common to all FC-4 types at that image pair.		

- **Word 0, Bit 15 - Originator process associator valid**

  0 = not meaningful

  1 = meaningful

- **Word 0, Bit 14 - Responder process associator valid**

  0 = not meaningful

  1 = meaningful

- **Word 0, Bit 13 - Establish image pair**

  0 = Exchange service parameters only

  1 = Establish image pair and exchange service parameters

- **Word 3, Bits 31–0 - Service parameters**

  Common service parameters are specified below. FC-4 TYPE code dependent service parameters shall be specified in the corresponding FC-4 document.

  No common service parameters are currently specified.

Reply link service sequence:

Link service reject (LS_RJT)

signifies rejection of the PRLI request

Accept (ACC)

signifies successful completion of the PRLI request

- Accept payload

The format of the accept payload is shown in table table A.5.

Table A.5 - PRLI accept payload

Item	**Size (Bytes)**
x‘02’ = ACC command code	1
x‘10’ = Page length	1
Payload length	2
Service parameter response pages	16 –max

Page length: Byte 1 of word 0 contains an 8-bit unsigned binary integer that specifies the length of each service parameter response page. The rightmost two bits shall be zeros.

Payload length: Bytes 2–3 of word 0 contain a 16-bit unsigned binary integer that specifies the length of the PRLI accept payload. The rightmost two bits shall be zeros. The value specified shall be greater than or equal to 20, and less than or equal to 65,532.

Service parameter response page(s): Words 1 through the maximum of the PRLI accept payload contain one or more service parameter response pages. Each service parameter response page contains service parameter responses for a single image pair or N_Port pair and is associated with a single FC-4 type or common to all FC-4 types at that image pair or N_Port pair.

The format of PRLI accept service parameter response pages is described in table A.6.

- **Word 0, Bits 31–24 - TYPE code or common service parameters**

  Identifies the protocol associated with this service parameter response page. If byte 0 of the first word of a service parameter response page contains the value x‘00’, the service parameter page contains service parameters common to all FC-4 types at that image pair or N_Port pair. If byte 0 of the first word of a service parameter response page contains a value other than x‘00’, the service parameter page contains service parameters for the FC-4 type indicated as specified by ANSI X3.230, FC-PH.

- **Word 0, Bits 23–16 - TYPE code extension**

  Reserved for future use.

- **Word 0, Bit 15 - Originator process associator valid**

  0 = not meaningful

  1 = meaningful

- **Word 0, Bit 14 - Responder process associator valid**

  0 = not meaningful

  1 = meaningful

- **Word 0, Bit 13 - Image pair established**

  Image pair established is valid only if bit 14 was set to one on the corresponding service parameter page

Table A.6 - PRLI accept service parameter response page format

Item	Word	Bit
Service parameter response page	0–3	31–0
TYPE code or common service parameters (1)	0	31–24
TYPE code extension	0	23–16
Originator process associator valid	0	15
Responder process associator valid	0	14
Image pair established	0	13
Reserved	0	12
Response code	0	11–8
Reserved	0	7–0
Originator process associator	1	31–0
Responder process associator	2	31–0
Service parameter response	3	31–0

NOTE 1 – If byte 0 of the first word of a service parameter page is set to the value hex’00’, the service parameter page is common to all FC-4 types at that image pair.

of the PRLI request.

0 = Image pair not established, see response code for additional information

1 = Image pair established

- **Word 0, Bits 11:8 - Response code**

  The response code field contains an encoded binary value indicating the result of the PRLI request. The meanings of the encoded response code values are shown in table A.7.

- **Word 3, Bits 31–0 - Service parameter response**

  Provides feedback to the originator as to the resultant state of the service parameters as returned by the responder.

A.2.3 PRLI/PRLO relationships

Process associator (PA) images can exist in the following relationships. Any of these relationships can be established as a default condition using mechanisms not specified by the standards.

A.2.3.1 PA not supported

If PAs are not supported by one or both of the originator and responder, no valid PA shall be specified in any frame of any exchange.

PRLI and PRLO commands shall have no more than a single page for the common service parameters and an additional single page for each additional type supported by the originator or responder. Each page shall indicate that both the originator PA and responder PA are invalid.

The PRLI/PRLO can be thought of as enabling and disabling the specified services for the entire N_Port.

Table A.7 - PRLI accept response code

Encoded Value Wd 1, Bits 11:8	Description
0000	Reserved
0001	Request executed
0010	The target image has no resources available for establishing image pairs between the specified source and destination N_Ports. The PRLI request may be retried.
0011	Initialization is not complete for the target image. The PRLI request may be retried.
0100	The target image corresponding to the responder PA specified in the PRLI request and PRLI accept response does not exist. The PRLI request shall not be retried.
0101	The target image has a predefined configuration which precludes establishing this image pair. The PRLI request shall not be retried.
0110	Request executed conditionally. Some service parameters were not able to be set to their requested state. See the service parameter response field for further details.
0111	The destination N_Port is unable to process multiple page PRLI requests. The PRLI request may be retried as a single page request.
1000:1111	Reserved

A.2.3.2 PA required by originator, supported by responder

If PAs are required by the originator, the originator is expected to communicate only when a valid PA is included in the initial association header. The responder shall provide a final association header and provide other association headers as required by ANSI X3.230, FC-PH. The PA establishes a routing to a particular host process. There is no relationship between PAs identified through one N_Port of a host system and another N_Port from the same host. Future Fibre Channel standards may define hunt and stripe groups using N_Port aliases. If so, then a process defined by a PA for the alias of the group shall be considered as the same process for all N_Ports of the group. The PA may additionally be used by the FC-4 in an FC-4 dependent manner.

The PRLI request pages shall have valid originator PA values and invalid responder PA values. This PRLI will inform the responder of the originator requirements and capabilities for each type of FC-4.

The PRLI and PRLO can be thought of as enabling and disabling the access of each originator process to the responder N_Port.

A.2.3.3 PA required by responder, supported by originator

If PAs are required by the responder, the responder is expected to communicate only when a valid responder PA is included in the initial association header. The responder shall provide a final association header and provide other association headers as required by ANSI X3.230, FC-PH. The PA establishes a routing to a particular responder process. There is no relationship between PAs identified through one N_Port of a responder and another N_Port from the same responder. Future Fibre Channel standards may define hunt and stripe groups using N_Port aliases. If so, then a process defined by a PA for the alias of the group shall be considered as the same process for all N_Ports of the group. The PA may additionally be used by the FC-4 in an FC-4 dependent manner.

The PRLI request pages shall have valid responder PA values and invalid originator PA values. The responder PA values shall be those obtained through an informative PRLI operation or by other methods not specified.

The PRLI and PRLO can be thought of as enabling and disabling the access of each responder process to the originating N_Port.

A.2.3.4 PA required by originator and responder

Communication can only take place between an originator and a responder when valid PAs are specified. Future Fibre Channel standards may define hunt groups and striping groups using N_Port aliases. If so, then a group of processes defined by a PA for the alias of the group shall be considered as the same group of processes for all N_Ports of the group. The PA may additionally be used by the FC-4 in an FC-4 dependent manner.

The PRLI request pages must have valid originator and responder PA values defined and must have the establish image pair bit set to one in order to create image pairs in an binding manner. Communication with image pairs that have not been established is not allowed, even if both PAs exist and have the proper requirements and capabilities.

PRLI and PRLO pages with complete originator PA and responder PA information enable or disable communication between the specified image pair. PRLI pages with incomplete originator PA and responder PA information are invalid.

A.2.4 Process login, mode of operation

PRLI has two modes of operation.

Nonbinding mode

Service parameter information is exchanged enabling subsequent negotiation for image pair establishment.

Binding mode

> Information is exchanged that explicitly establishes a relationship between processes in the communicating N_Ports. The relationship does not allow any communication types or paths other than those established by the PRLI.

The use of a binding PRLI page requires that the originator have precise and detailed knowledge of the PAs and capabilities available in the responder. That information may be obtained from Directory Services, implicitly from configuration information obtained outside the scope of FC, or by performing a nonbinding PRLI.

Binding or nonbinding mode is determined by the setting of the establish image pair bit in the PRLI request page.

The service parameters included in a page may be either requirements or capabilities. Capabilities indicate those FC-4 properties that describe the role and state of the node in the FC-4. Requirements indicate those FC-4 properties that must be agreed upon by both nodes for operation with an FC-4. See 6.1, “Overview of process login/logout,” for examples of capabilities and requirements in FCP.

A.2.5 Process login, protocol

A.2.5.1 PA required by originator and responder

For each PRLI request page, the originator and responder PA validity bits are set to one and valid originator and responder PAs are specified. Service parameter requirements are set to those that correspond to the combined understanding expected to be agreed to by the responder, given the knowledge that the originator PA has of the responder PA. Service parameter capabilities shall be those of the originator only. The specification of service parameters as either requirements or capabilities is specified in the corresponding FC-4 document.

For each ACC response page, the originator and responder PA validity bits are set to one and valid originator and responder PAs are specified. Service parameter requirements are set to those that are agreed to by the responder. Service parameter capabilities shall be those of the responder only. Error responses are possible.

Each page identifies a binding mode PRLI operation between one image at the originator N_Port and one image at the responder N_Port.

A.2.5.2 PA required by originator, supported by responder

For each PRLI request page, the originator PA validity bit is set to one and the responder PA validity bit is set to zero. A valid originator PA is specified. Service parameter requirements are set to those that correspond to the combined understanding expected to be agreed to by the responder, given the knowledge that the originator PA has of the responder PA. Service parameter capabilities shall be those of the originator only. The specification of service parameters as either requirements or capabilities is specified in the corresponding FC-4 document.

For each ACC response page the originator PA validity bit is set to one and the responder PA validity bit is set to zero. The originator PA is returned. Service parameter requirements are set to those that are agreed to by the responder. Service parameter capabilities shall be those of the responder only. Error responses are possible.

Each page identifies a binding mode PRLI operation between one image at the originator N_Port and the responder N_Port.

A.2.5.3 PA supported by originator, required by responder

For each PRLI request page, the originator PA validity bit is set to zero and the responder PA validity bit is set to one. A valid responder PA is specified. Service parameter requirements are set to those that correspond to the combined understanding expected to be agreed to by the responder, given the knowledge that the originator PA has of the responder PA. Service parameter capabilities shall be those of the originator only. The specification of service parameters as either requirements or capabilities is specified in the corresponding FC-4 document.

For each ACC response page, the originator PA validity bit is set to zero and the responder PA validity bit is set to one. The responder PA is returned. Service parameter requirements are set to those that are agreed to by the responder. Service parameter capabilities shall be those of the responder only. Error responses are possible.

Each page identifies a binding mode PRLI operation between the originator N_Port and one image at the responder N_Port.

A.3 Process logout

The process logout (PRLO) extended link service request shall be used to request invalidation of the operating environment between an image at the initiating N_Port and an image at the recipient N_Port. PRLO frees resources committed by a previous PRLI function. ULP behavior following successful execution of the PRLO function is specified in the corresponding FC-4 document.

Examples of PRLO usage include image re-configuration.

One or more image pairs is removed with a single PRLO request and accept reply sequence set. Other image pairs associated with the same or different N_Ports are unaffected. After an image pair is removed, IUs may not be sent or received using that image pair.

A ULP may choose to remove multiple image pairs with a single PRLO request. This can be accomplished by specifying multiple logout parameter pages within a single PRLO request. However, a destination N_Port may be unable to execute a PRLO request containing multiple logout parameter pages. No protocol to determine the capability of a PRLO responder to execute PRLO requests containing multiple logout parameter pages is specified. An accept response containing a nonzero response code is returned in the case that a PRLO responder is unable to execute a PRLO request containing multiple logout parameter pages.

If a PRLO request is received for an image pair that does not exist, the request is accepted, provided that no link errors are detected, and the accept response is sent.

If a link error is detected when a PRLO request is received, the sequence is discarded, and the appropriate response, if any, for the link error recognized is sent. The N_Port originating the PRLO request shall not consider an image pair to be removed until it receives an error-free accept reply sequence. The N_Port responding to the PRLO request shall not consider an image pair to be removed until the accept reply sequence is completely sent without error. An N_Port that receives a link-busy or link-reject reply in response to a PRLO request may retry the PRLO request. The number of retries is system dependent.

Unless the requesting N_Port receives a valid response to a PRLO request, that N_Port cannot assume that the requested action has or has not taken place. If an invalid response is received by an N_Port to the PRLO request, the N_Port can retry the request. The number of retries is system dependent.

Protocol:

Process logout request sequence

Accept reply sequence

Format:

FT-1

Addressing:

The S_ID field designates the N_Port requesting process logout. The D_ID field designates the destination N_Port of the process logout.

Payload:

The format of the payload is shown in table A.8.

Table A.8 - PRLO payload

Item	Size (Bytes)
x‘21’ = PRLO command code	1
x‘10’ = Page length	1
Payload length	2
Logout parameter pages	16 – max

Page length:

Byte 1 of word 0 contains an 8-bit unsigned binary integer that specifies the length of each logout parameter page. The rightmost two bits shall be zeros.

Payload length:

Bytes 2–3 of word 0 contain a 16-bit unsigned binary integer that specifies the length of the PRLO payload. The rightmost two bits shall be zeros. The value specified shall be greater than or equal to 20, and less than or equal to 65,532.

Logout parameter page(s):

Words 1 to the maximum of the PRLO payload contain one or more logout parameter pages. Each logout parameter page contains logout parameters for a single image pair and is associated with all FC-4 TYPE codes supported by that image pair.

The format of PRLO logout parameter pages is described in table A.9.

Table A.9 - PRLO logout parameter page format

Item	Word	Bit
Logout parameter page	0–3	31–0
Reserved	0	31–16
Originator process associator valid	0	15
Responder process associator valid	0	14
Reserved	0	13–0
Originator process associator	1	31–0
Responder process associator	2	31–0
Reserved	3	31–0

- **Word 0, Bit 15 - Originator process associator valid**

  0 = not meaningful

  1 = meaningful

- **Word 0, Bit 14 - Responder process associator valid**

  0 = not meaningful

  1 = meaningful

Reply link service sequence:

Link service reject (LS_RJT)

signifies rejection of the PRLO request

Accept (ACC)

signifies successful completion of the PRLO request

- Accept payload

The format of the accept payload is shown in table A.10.

Page length:

Table A.10 - PRLO accept payload

Item	Size (Bytes)
x‘02’ = ACC command code	1
x‘10’ = Page length	1
Payload length	2
Logout parameter response pages	16 –max

Byte 1 of word 0 contains an 8-bit unsigned binary integer that specifies the length of each logout parameter response page. The rightmost two bits shall be zeros.

Payload length:

Bytes 2–3 of word 0 contain a 16-bit unsigned binary integer that specifies the length of the PRLO accept payload. The rightmost two bits shall be zeros. The value specified shall be greater than or equal to 20, and less than or equal to 65,532.

Logout parameter response page(s):

Words 1 to the maximum of the PRLO accept payload contain one or more logout parameter response pages. Each logout parameter response page contains a logout parameter response for a single image pair and is associated with all FC-4 Type codes supported by that image pair.

The format of PRLO accept logout parameter response pages is described in table A.11.

Table A.11 - PRLO accept logout parameter response page format

Item	Word	Bit
Logout parameter response page	0–3	31–0
Reserved	0	31–16
Originator process associator valid	0	15
Responder process associator valid	0	14
Reserved	0	13–12
Response code	0	11–8
Reserved	0	7–0
Originator process associator	1	31–0
Responder process associator	2	31–0
Reserved	3	31–0

- **Word 0, Bit 15 - Originator process associator valid**

  0 = not meaningful

  1 = meaningful

- **Word 0, Bit 14 - Responder process associator valid**

  0 = not meaningful

1 = meaningful

- **Word 0, Bits 11:8 - Response code**

  The response code field contains an encoded binary value indicating the result of the PRLO request and the status of the image pair. The meanings of the encoded response code values are shown in table A.12.

Table A.12 - PRLO accept response code

Encoded Value Wd 1, Bits 11:8	Description
0000	Reserved
0001	Request executed
0010:0011	Reserved
0100	The target image corresponding to the responder PA specified in the PRLO request and PRLO accept response does not exist. The PRLO request shall not be retried.
0101:0110	Reserved
0111	The destination N_Port is unable to process multiple page PRLO requests. The PRLO request may be retried as a single page request.
1000:1111	Reserved

A.3.1 PRLO Operation

PRLO operates by referencing particular PA image pairs. Pages can be mixed in any combination in a PRLO request. The PRLO accept is required to present a response page for every request page. Pages may be individually marked as being in error.

A PRLO page identifies a particular image pair to logout by marking either or both the originator PA and the responder PA as valid. Only that image pair is logged out. No further communication for the affected FC-4(s) is possible between these two images. It shall be the responsibility of the ULPs to insure that all active operations over an image pair have been orderly and properly terminated prior to issuing a PRLO request. Following PRLO execution, all active sequences shall be terminated by invoking abort sequence protocol (ABTS). Following the completion of ABTS protocol, all active exchanges shall be terminated by invoking the Abort exchange (ABTX) protocol or the FCP recovery abort. Operations and states for other image pairs are not affected.

ULP attempts to communicate over an image pair that has not been established or has been abnormally terminated shall be acknowledged in the normal manner. The originator may then perform a PRLO operation for the affected image pair in order to properly terminate the operating environment at both the originator and responder.

Annex B
FCP examples

(informative)

B.1 Examples of the use of FCP information units

The following sections provide examples of the use of FCP information units. The functions enclosed in square brackets summarize actions that are not specified by the FCP document, but are typically executed by SCSI initiators and targets. Sequence streaming may optionally be performed between any two IUs that do not transfer sequence initiative.

B.1.1 SCSI FCP read operation

A typical SCSI FCP read operation with a single data IU and using FCP_XFER_RDY is shown in table B.1.

Table B.1 - FCP read operation, example

Initiator function	IU	Target function
Command request	T1, FCP_CMND ->	
		[Prepare data transfer]
	<- I2, FCP_XFER_RDY	Data delivery request
	<- I3, FCP_DATA	Data in action
		[Prepare response message]
	<- I4, FCP_RSP	Response
[Indicate command completion]		

B.1.2 SCSI FCP write operation

A typical SCSI FCP write operation with three data IUs and using FCP_XFER_RDY is shown in table B.2.

Table B.2 - FCP write operation, example

Initiator function	IU	Target function
Command request	T1, FCP_CMND ->	
		[Prepare data transfer buffer]
	<- I1, FCP_XFER_RDY	First data delivery request
First Data Out Action	T6, FCP_DATA ->	
	<- I1, FCP_XFER_RDY	Second data delivery request
Second Data Out Action	T6, FCP_DATA ->	
	<- I1, FCP_XFER_RDY	Last data delivery request
Last Data Out Action	T6, FCP_DATA ->	
		[Prepare response message]
	<- I4, FCP_RSP	Response
[Indicate command completion]		

B.1.3 SCSI FCP operation with no data transfer or with check condition

A typical SCSI FCP operation terminating without data transfer, either because of an error or because the SCSI command does not require any data transfer, is shown in table B.3.

Table B.3 - FCP operation without data transfer, example

Initiator function	IU	Target function
Command request	T1, FCP_CMND ->	
		[perform command]
	<- I4, FCP_RSP	Response
[Indicate command completion]		

B.1.4 SCSI FCP read operation FCP_XFR_RDY disabled

A typical SCSI read operation performed without the use of FCP_XFER_RDY is shown in table B.4. Multiple FCP_DATA IUs are sent in this example.

Table B.4 - FCP read operation with FCP_XFER_RDY disabled, example

Initiator function	IU	Target function
Command request	T1, FCP_CMND ->	
		[Prepare data transfer]
	<- I3, FCP_DATA	Data in action
	<- I3, FCP_DATA	Data in action
	<- I3, FCP_DATA	Data in action
		[Prepare response message]
	<- I4, FCP_RSP	Response
[Indicate command completion]		

B.1.5 SCSI FCP write operation with FCP_XFR_RDY disabled

A typical SCSI write operation performed with FCP_XFER_RDY disabled is shown in table B.5. Only the first transfer is performed without a requesting FCP_XFER_RDY.

Table B.5 - FCP write operation with FCP_XFER_RDY disabled, example

Initiator function	IU	Target function
Command request	T2, FCP_CMND ->	
Data Out Action	T6, FCP_DATA ->	
	<- I1, FCP_XFER_RDY	Second data delivery request
Data Out Action	T6, FCP_DATA ->	
	<- I1, FCP_XFER_RDY	Last data delivery request
Data Out Action	T6, FCP_DATA ->	
		[Prepare response message]
	<- I4, FCP_RSP	Response
[Indicate command completion]		

B.1.6 SCSI linked commands

A SCSI write operation linked to a SCSI read operation is shown in table B.6. The commands are using the FCP_XFER_RDY IU. Intermediate Status in the FCP_RSP, together with the link control bits present in the CDB of the FCP_CMND indicate that the second operation is linked to the first.

Table B.6 - FCP linked commands, example

Initiator function	IU	Target function
Command request	T1, FCP_CMND ->	
		[Prepare data transfer]
	<- I2, FCP_XFER_RDY	Data delivery request
	<- I3, FCP_DATA	Data in action
		[Prepare response message]
	<- I5, FCP_RSP	Response
[Perform command linking]		
Command request	T3, FCP_CMND ->	
		[Prepare data transfer buffer]
	<- I1, FCP_XFER_RDY	Data delivery request
Data Out Action	T6, FCP_DATA ->	
		[Prepare response message]
	<- I4, FCP_RSP	Response
[Indicate command completion]		

B.1.7 SCSI FCP read operation with combined data in and response

An example of a SCSI read operation using a combined data in and response IU is shown in table B.7. If more than one FCP_DATA IU is transmitted, the response is only combined for the last Data In.

Table B.7 - FCP read operation with combined data and response, example

Initiator function	IU	Target function
Command request	T1, FCP_CMND ->	
		[Prepare data transfer]
	<- I2, FCP_XFER_RDY	Data delivery request
	<- I6, FCP_DATA and FCP_RSP	Data in action with response
[Indicate command completion]		

B.1.8 SCSI FCP write operation with combined command and data out

An example of a SCSI write operation using a combined Command and Data Out IU is shown in table B.8. If more than one FCP_DATA IU is transmitted, only the first is combined with the FCP_CMND. FCP_XFER_RDY must be disabled to allow this type of operation. Appropriate buffering in the target must be guaranteed for such operations to be successful.

Table B.8 - FCP write operation with combined command and data out, example

Initiator function	IU	Target function
Command request combined with Data Out	T8, FCP_CMND and -> FCP_DATA	
		[Prepare response message]
	<- I4, FCP_RSP	Response
[Indicate command completion]		

B.1.9 SCSI FCP task management function

An example of a SCSI Task Management function is shown in table B.9. Additional link services may be required in some cases to complete the activities initiated by the Task Management function.

Table B.9 - FCP task management function, example

Initiator function	IU	Target function
Command request, no CDB	T1, FCP_CMND ->	
		[Do Task Management]
	<- I4, FCP_RSP	Response
[Indicate task management complete]		

B.2 FCP write example, frame level

A chart of the sequences and frames typically transmitted to perform a FCP write is shown in figure B.1. All frames of a sequence have a frame level FC-PH acknowledgment returned automatically as part of the link control.

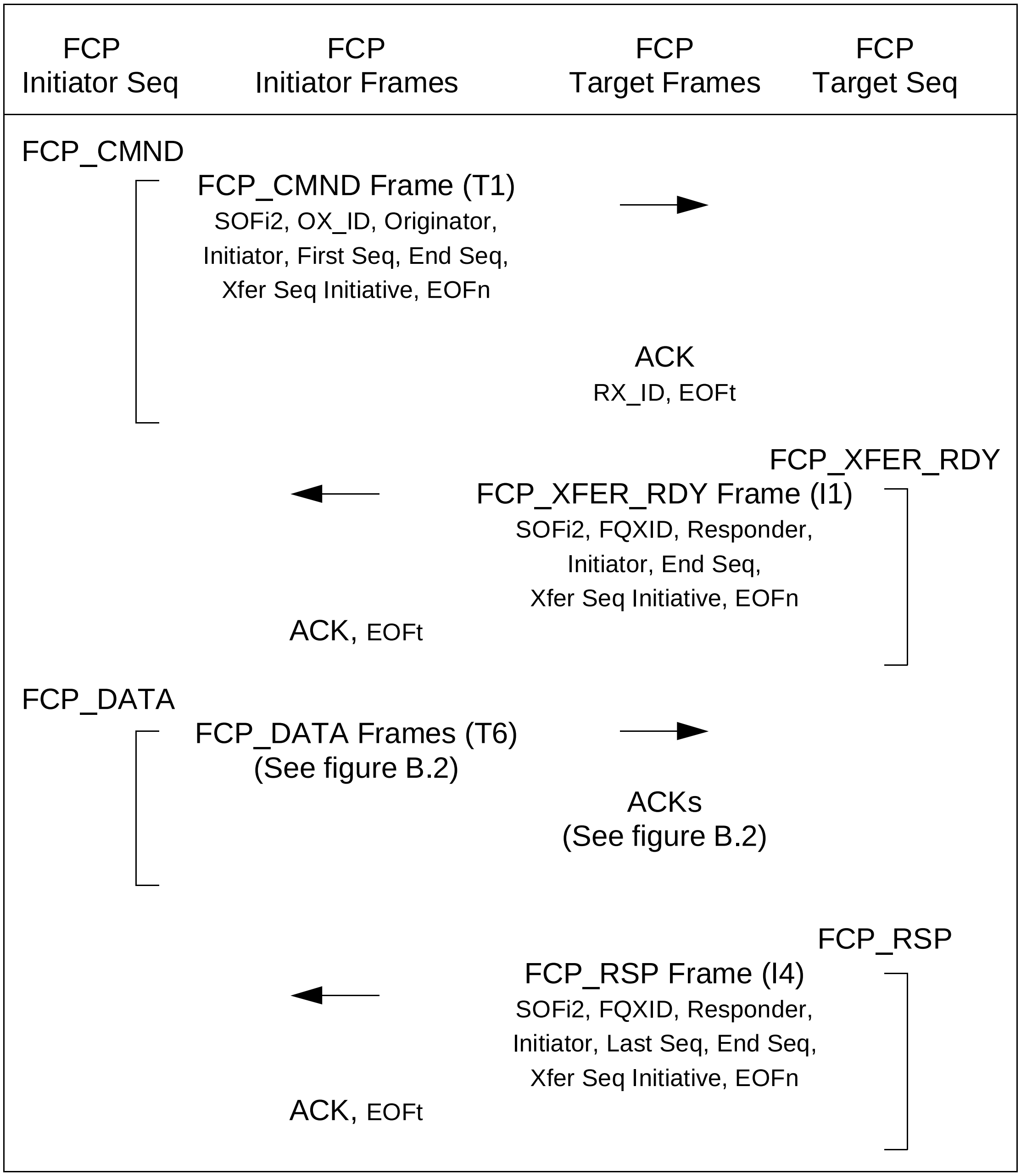

Figure B.1 - Example of class 2 FCP write I/O operation

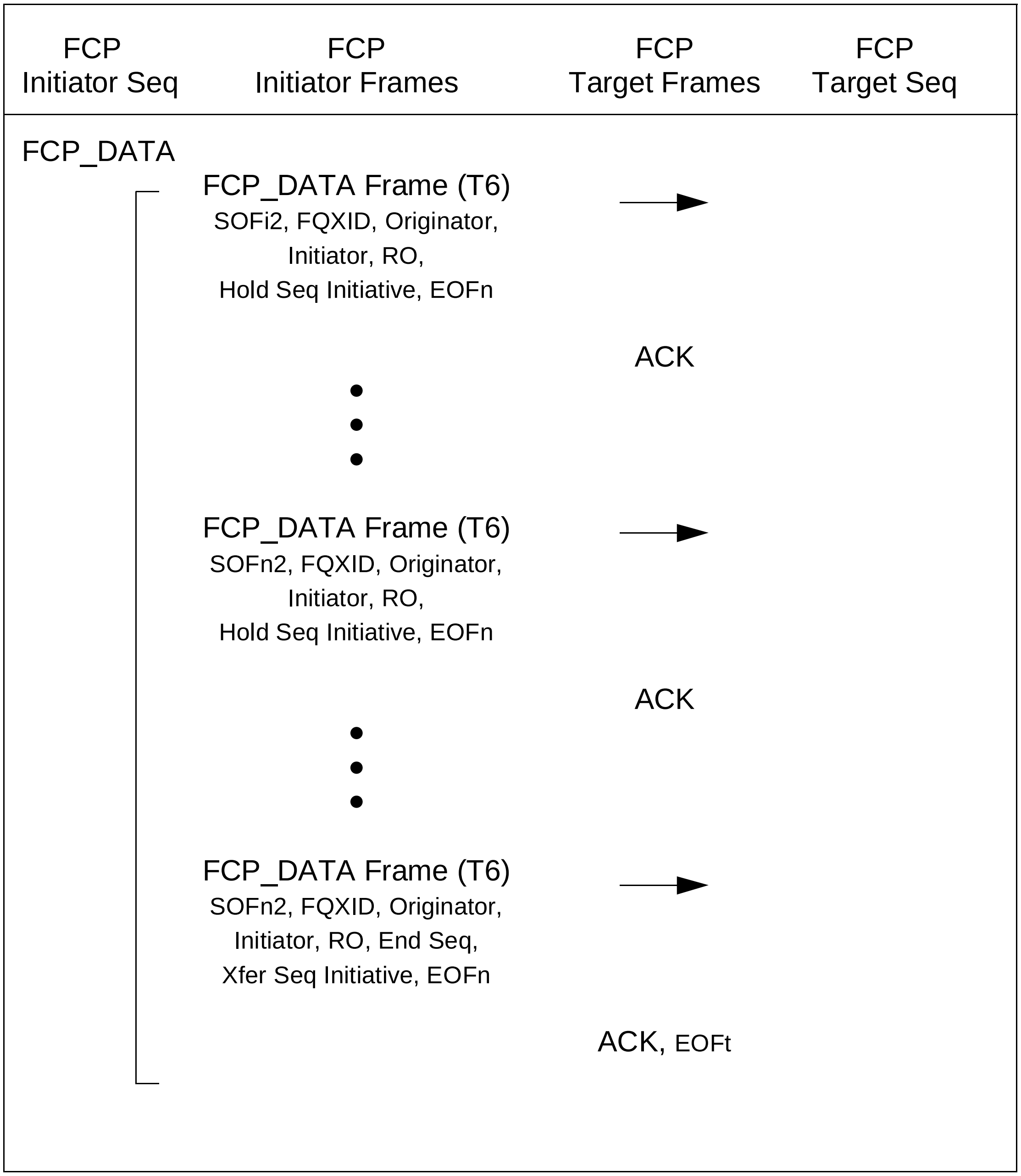

Figure B.2 - Example of class 2 FCP_DATA write sequence

B.3 FCP read example, frame level

A chart of the sequences typically transmitted to perform a FCP read is shown in figure B.3. All frames of a sequence have a frame level FC-PH acknowledgment returned automatically as part of the link control.

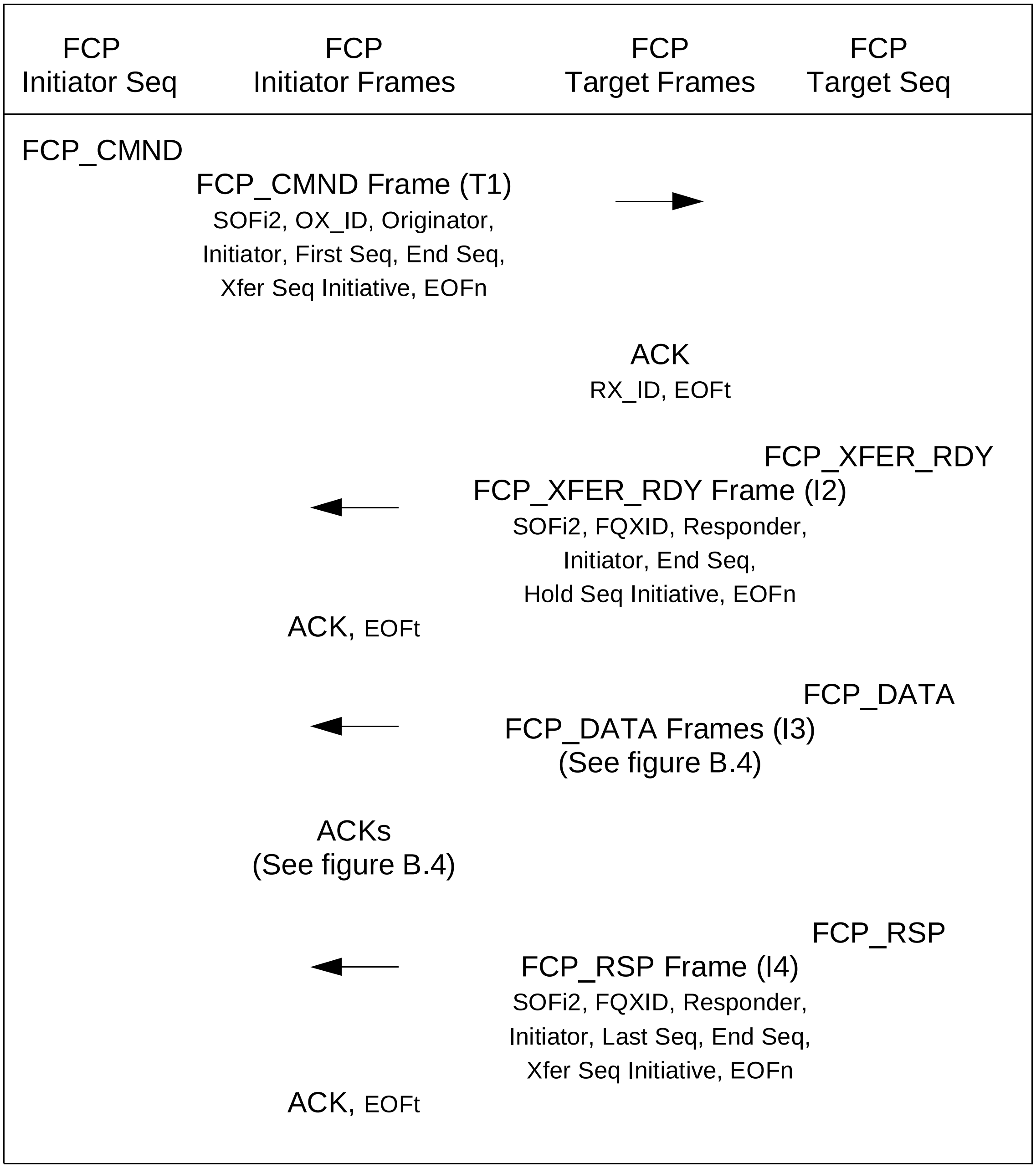

Figure B.3 - Example of class 2 FCP read I/O operation

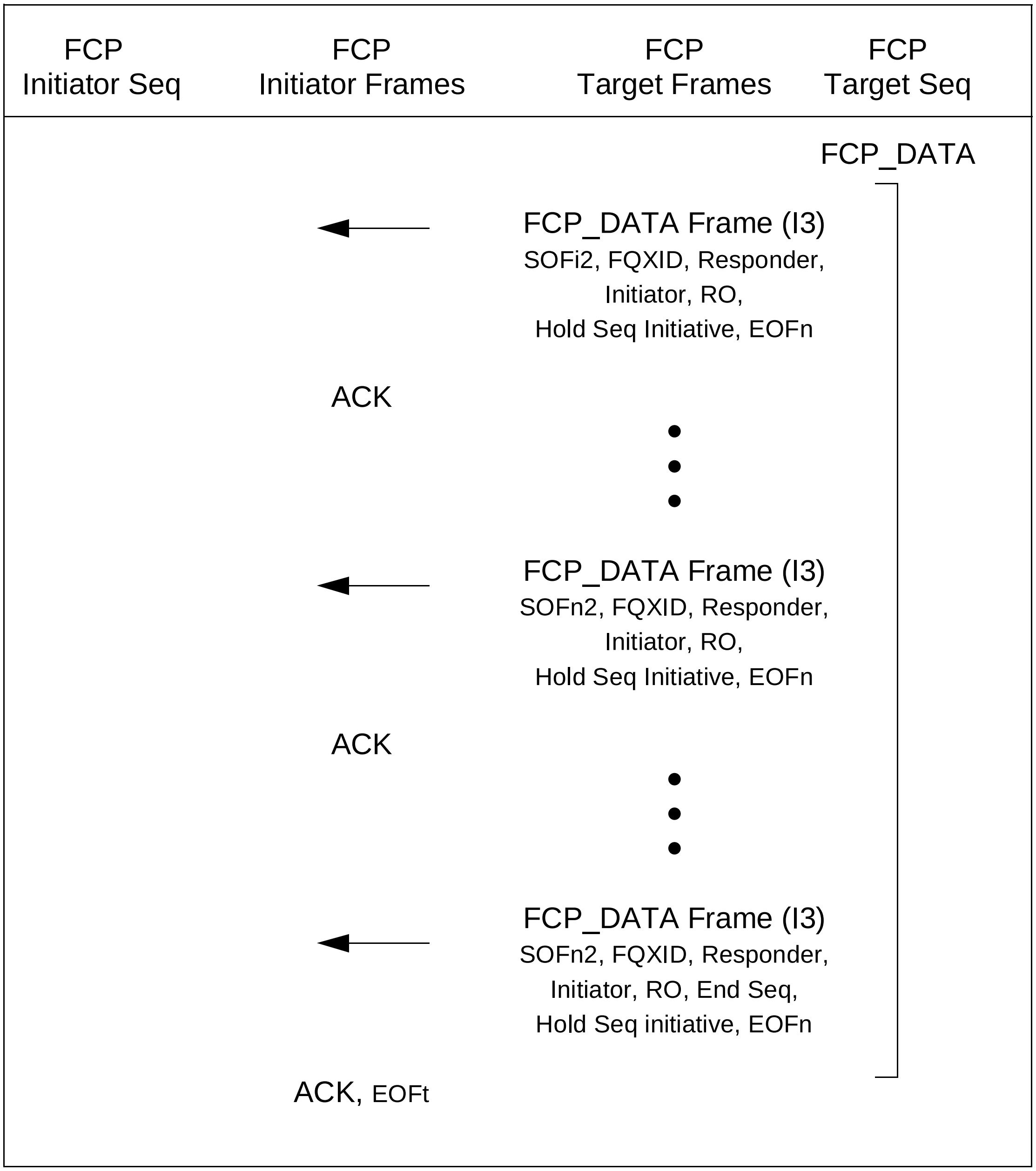

Figure B.4 - Example of class 2 FCP_DATA read sequence

Annex C
Logical Unit Number example

(informative)

C.1 Logical Unit Number definition

The 8-byte FCP_LUN (section 7.1.1 on page 24) is used to address the proper logical unit. The actual format of the FCP_LUN depends on the model of the device at the FCP_Port. An example of the use of the logical unit number, based on the addressing model contained within the SCSI-3 Controller Commands Standard (dpANS ANSI X3T10/1047D), is provided in the following sections.

C.2 Definition of SCSI-3 Controller Commands (SCC) addressing model

The logical unit number structure defined for the SCC addressing model allows up to four levels of SCSI-3 storage array devices to be addressed and defined. The addressing structure allows the addressing at each level in the hierarchy of an SCSI-3 storage array, all peripheral device logical unit numbers, and all volume set logical unit numbers.

The format of the logical unit number for this example is shown in table C.1.

Table C.1 - FCP_LUN format, SCSI-3 storage array example

Field	Size
FCP_LUN_0 (first level)	2 bytes
FCP_LUN_1 (second level)	2 bytes
FCP_LUN_2 (third level)	2 bytes
FCP_LUN_3 (fourth level)	2 bytes

Each level of the addressing has the addressing structure show in table C.2.

Table C.2 - FCP_LUN_X format, SCSI-3 storage array example

Bit Byte	7	6	5	4	3	2	1	0
n	Address_Method		LUN/Bus Number					
n+1	LUN/Target							

A notation which can be used to write these addresses is based on the values in each of the fields.

Generalized Address: (M0,B0,T0):(M1,B1,T1):(M2,B2,T2):(M3,B3,T3)

Where M = Address_Method
P = Peripheral Device
V = Volume Set

B = SCSI Bus within the SCSI-3 Storage Array
0 = Reserved for SCSI-3 storage array and internal SCSI-3 storage array functions
1–63 = SCSI Bus identifier

T = SCSI Target ID on the specified SCSI Bus
0–256 = SCSI ID on specified SCSI Bus

and B and T are used as a single LUN to identify the address of a volume set.

If a logical unit is identified by a certain address, those levels of the hierarchy lower than the logical unit are not utilized and are ignored. The address is structured such that as each level of the hierarchy is analyzed, the first level can be removed, the addresses shifted left by one level, and the remaining levels will have an identical and normal addressing structure. The Address_Method bits have the values allowed in table C.3.

Table C.3 - Address_Method values

Address_Method	**Bit**	
	7	**6**
Peripheral device logical unit/SCSI-3 storage array	0	0
Volume set logical unit	0	1
reserved	1	0
reserved	1	1

The SCSI-3 storage array physical configuration used in the following examples is shown in figure C.1.

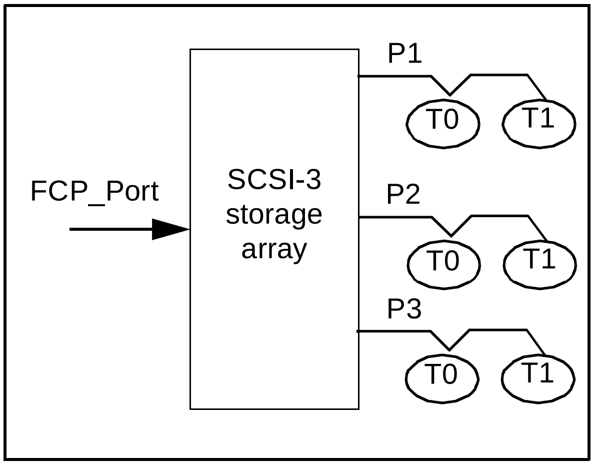

Figure C.1 - SCSI-3 storage array example

C.3 Use of logical unit number in a SCSI-3 storage array.

At each level of the hierarchy, the SCSI-3 storage array can be addressed, each of the peripheral devices can be addressed, or volume sets constructed from the peripheral devices may be addressed. Examples of the use of the logical unit number in a SCSI-3 storage array device are shown below.

C.3.1 Addressing of SCSI disk array (SCSI-3 storage array), Example:

It is required by the FCP that the FCP_Port shall be able to send a successful INQUIRY command to logical unit with the logical unit number of all zeros. That address (‘0000 0000 0000 0000’ hexadecimal) corresponds to the first level SCSI-3 storage array address of:

(P,0,0):(U,U,U):(U,U,U):(U,U,U)

Because the SCSI-3 storage array’s address is identified in the first level of the hierarchy, the remaining levels are unused (U).

The view of the SCSI-3 storage array device when the first level is addressed is shown in figure C.2.

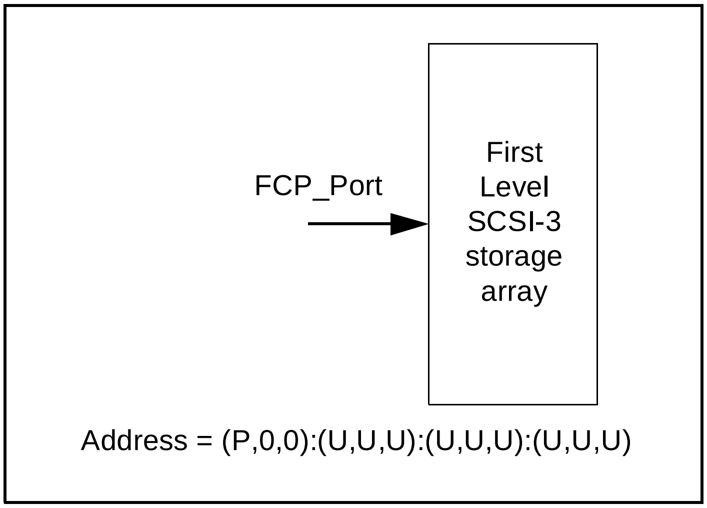
FCP_Port
First Level SCSI-3 storage array
Address = (P,0,0):(U,U,U):(U,U,U):(U,U,U)

Figure C.2 - SCSI-3 storage array addressing

C.3.2 Addressing of volume set logical unit: example

A SCSI-3 storage array may combine a large number of individual peripheral devices into the logical image of one or more disk drive volume sets. Volume sets managed by the first level SCSI-3 storage array are addressed in the following manner:

(V, Addr Hi, Addr Lo):(U,U,U):(U,U,U):(U,U,U)

An example of a SCSI-3 storage array image with two large logical units is shown in figure C.3.

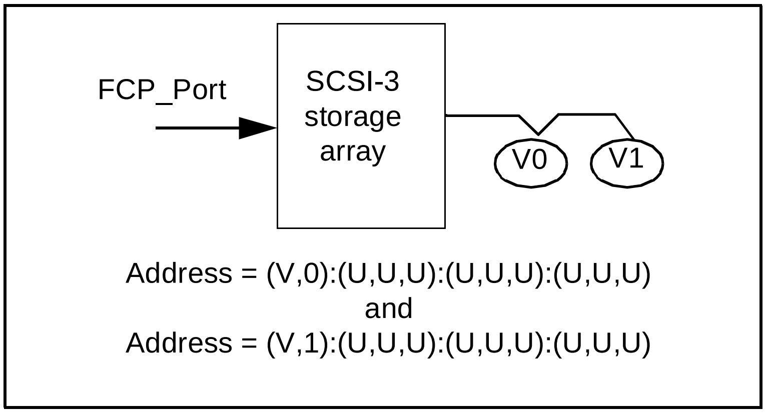
FCP_Port
SCSI-3 storage array
V0
V1
Address = (V,0):(U,U,U):(U,U,U):(U,U,U)
and
Address = (V,1):(U,U,U):(U,U,U):(U,U,U)

Figure C.3 - SCSI-3 storage array unit addressing

C.3.3 Addressing of peripheral device logical unit: example

A maintenance utility may require access to individual logical units within a disk array to perform diagnostic operations, microcode updates of attached devices, and other functions. Peripheral device logical units managed by the first level SCSI-3 storage array are addressed in the following manner:

(P,B,T):(U,U,U):(U,U,U):(U,U,U)

Standard single LUN targets are understood to have a LUN value of zero. If the individual disk has more than one LUN, that is identified as a peripheral device logical unit with a Bus value of zero in the next level.

An example of peripheral device logical unit addressing is shown in figure C.4.

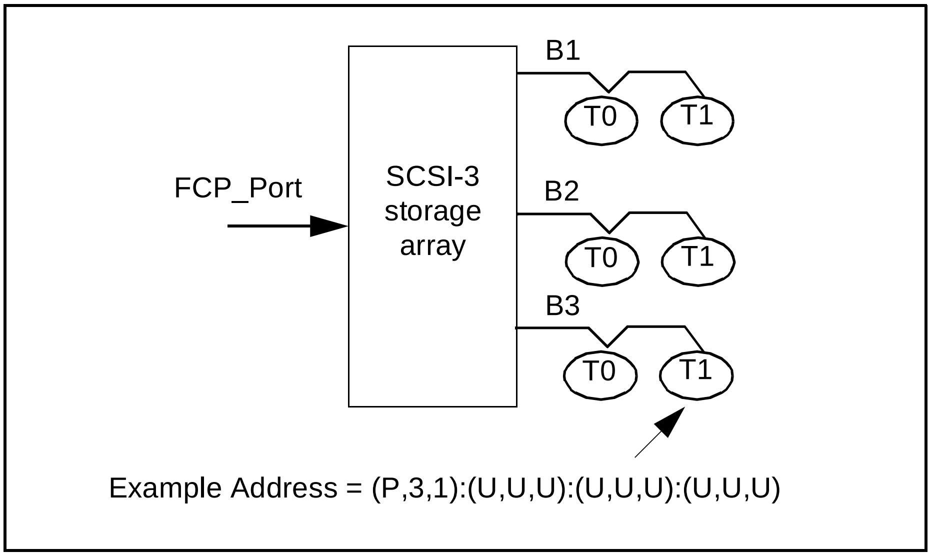

Figure C.4 - SCSI-3 storage array diagnostic addressing

Annex D
Bibliography

(informative)

The following draft standards are references used throughout this document. They are all in the final stages of approval, but are not approved as of this edition of the FCP standard.

FC-AL

draft proposed ANS ANSI X3T11/960D, Revision 4.4, *Fibre Channel Arbitrated Loop (FC-AL)*

FC-SB

draft proposed ANS ANSI X3T11/957D, Revision 3.1, *Fibre Channel Single Byte Command Sets (SBCCS) Mapping Protocol (FC-SB)*

CAM

draft proposed ANS ANSI X3.232-199X, Revision 4, *Information technology - SCSI Common Access Method*

SCC

draft proposed ANS ANSI X3T10/1047D, Revision 4, *Information Technology - SCSI-3 Controller Commands (SCC)*

SPC

draft proposed ANS ANSI X3T10/995D, Revision 6, *Information Technology - SCSI-3 Primary Commands (SPC)*

Reference is made to these documents by the abbreviation listed above in bold face.