Copies of this document may be purchased from:
Global Engineering, 15 Inverness Way East,
Englewood, CO 80112-5704
Phone: (800) 854-7179 or (303) 792-2181 Fax: (303) 792-2192

INCITS 548-201x

FIBRE CHANNEL

GENERIC SERVICES - 8

(FC-GS-8)

REV 11.06

INCITS working draft proposed
American National Standard
for Information Technology

April 10, 2019

Secretariat: Information Technology Industry Council

NOTE:

This is a working draft American National Standard of Accredited Standards Committee INCITS. As such this is not a completed standard. The T11 Technical Committee or anyone else may modify this document as a result of comments received anytime, or during a future public review and its eventual approval as a Standard. Use of the information contained herein is at your own risk.

POINTS OF CONTACT:

Steve Wilson (T11 Chair)
Broadcom ~~Limited~~
130 Holger Way
San Jose, CA 95134
Voice: 408-333-8128
steve.wilson@broadcom.com

Craig W. Carlson (T11 Vice Chair)
Qlogic Corporation
12701 Whitewater Drive
Minnetonka, MN 55343
Voice: 952-687-2431
craig.carlson@qlogic.com

Craig W. Carlson (T11.3 Chair)
QLogic Corporation
12701 Whitewater Drive
Minnetonka, MN 55343
Voice: 952-687-2431
craig.carlson@qlogic.com

David Peterson (FC-GS-8 Chair)
Broadcom ~~Limited~~
1230 Northland Dr
Mendota Heights, MN 55120
Voice: 763-248-9374
david.peterson@broadcom.com

Jason Rusch (FC-GS-8 Editor)
Viavi Solutions Inc.
6001 America Center Drive, 6th Floor
San Jose, CA 95002
Voice: 240-404-1148
jason.rusch@viavisolutions.com

Release Notes for version 11.06

- Incorporated T11-2019-00081-v001 - Updating EUI-64 and Company_ID text in FC-GS-8.

Release Notes for version 11.05

- Incorporated T11-2018-00329-v000 - Application Server Entity additions / updates.
- Incorporated T11-2019-00012-v002 - Port Speed Capabilities / Port Operating Speed changes.

Release Notes for version 11.04

- Incorporated T11-2018-00100-v001 - FDMI Register Port clarification.
- Incorporated T11-2018-00102-v003 - Application Server additions.
- Incorporated T11-2018-00102-v004 - Application Server update.
- Incorporated T11-2018-00104-v001 - Printable ASCII String clarification.
- Incorporated T11-2018-00121-v003 - Fabric_Name usage.
- Incorporated T11-2018-00166-v000 - FDMI Register Port update.

Release Notes for version 11.03:

- Editorial changes specified in the April 2017 meeting minutes.
- Incorporated T11-2017-00309-v001 - Fabric Configuration Server Port Speed Object.
- Incorporated T11-2017-00313-v000 - FC-4 types description.

Release Notes for version 11.02:

- Added clause references to command tables.
- Incorporated 16-344v0 - Get Alias Names.
- Incorporated 16-489v0 - Name Value Characters.
- Incorporated T11-2017-00025-v002 - base application identifier description.
- Incorporated T11-2017-00026-v001 - rename EFCS commands.

Release Notes for version 11.01:

- Incorporated 15-433v1 - VE Identification Server.
- Incorporated 16-007v3 - Application Services.
- Incorporated 16-204v0 - VE Identification Server Additions.

Release Notes for version 11.00:

- Initial revision of FC-GS-8.

BSR INCITS 548-201x

American National Standard
for Information Technology

Fibre Channel —
Generic Services - 8 (FC-GS-8)

Secretariat

Information Technology Industry Council

Approved (not yet approved)

American National Standards Institute, Inc.

Abstract

This standard specifies Generic Services that may be used to support management and operation of a Fibre Channel Fabric. It incudes Services relating to device and topology discovery, Fabric Zoning, Fibre Channel security, time synchronization, address alias groups, event reporting, and performance monitoring.

This standard provides clarifications, corrections, Service enhancements, and new Services to the specification in INCITS 427:2007, and is recommended rather than INCITS 427:2007 to guide the development of new products. In addition, this standard removes the specifications of certain functions no longer of significance to the community.

American National Standard

Approval of an American National Standard requires review by ANSI that the requirements for due process, consensus, and other criteria for approval have been met by the standards developer.

Consensus is established when, in the judgement of the ANSI Board of Standards Review, substantial agreement has been reached by directly and materially affected interests. Substantial agreement means much more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered, and that a concerted effort be made towards their resolution.

The use of American National Standards is completely voluntary; their existence does not in any respect preclude anyone, whether he has approved the standards or not, from manufacturing, marketing, purchasing, or using products, processes, or procedures not conforming to the standards. The American National Standards Institute does not develop standards and will in no circumstances give an interpretation of any American National Standard. Moreover, no person shall have the right or authority to issue an interpretation of an American National Standard in the name of the American National Standards Institute. Requests for interpretations should be addressed to the secretariat or sponsor whose name appears on the title page of this standard.

CAUTION NOTICE: This American National Standard may be revised or withdrawn at any time. The procedures of the American National Standards Institute require that action be taken periodically to reaffirm, revise, or withdraw this standard. Purchasers of American National Standards may receive current information on all standards by calling or writing the American National Standards Institute.

CAUTION: The developers of this standard have requested that holders 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, following calls for the identification of patents that may be required for the implementation of the standard, notice of one or more such claims has been received. By publication of this standard, no position is taken with respect to the validity of this claim or of any rights in connection therewith. The known patent holder(s) has (have), however, filed a statement of willingness to grant a license under these rights on reasonable and nondiscriminatory terms and conditions to applicants desiring to obtain such a license. Details may be obtained from the publisher.

Published by

American National Standards Institute
11 West 42nd Street, New York, NY 10036

Printed in the United States of America

~~Foreword~~ ~~(This Foreword is not part of American National Standard INCITS 548-201x.)~~

~~The Fibre Channel Generic Services (FC-GS-8) standard describes in detail Generic Services introduced in FC-FS-4. In addition, this document describes any ancillary functions and Services required to support Generic Services.~~

~~This standard was developed by Task Group T11 of Accredited Standards Committee INCITS during 2006-2009. The standards approval process started in 2008. This document includes annexes that are informative and are not considered part of the standard.~~

~~Requests for interpretation, suggestions for improvements or addenda, or defect reports are welcome. They should be sent to the INCITS Secretariat, Information Technology Industry Council, 1250 Eye Street, NW, Suite 200, Washington, DC 20005-3922.~~

~~This standard was processed and approved for submittal to ANSI by the International Committee for Information Technology Standards (INCITS). Committee approval of the standard does not necessarily imply that all committee members voted for approval.~~

At the time it approved this standard, INCITS had the following members:

(to be filled in by INCITS)

Technical Committee T11 on Lower Level Interfaces, which reviewed this standard, had the following members:

Steve Wilson, Chair
~~Claudio DeSanti, Vice-Chair~~
~~Richard Johnson, Secretary~~

T11 Voting Members:

Company Name

(to be filled in prior to submission to INCITS)

Task Group T11.3 on Fibre Channel Protocols, which developed and reviewed this standard, had the following members:

Craig W. Carlson, Chair
~~Louis Ricci, Vice-Chair~~
~~Landon Noll, Secretary~~

T11.3 Voting Members:

Company Name

(to be filled in prior to submission to INCITS)

Introduction

FC-GS-8 is one of the Fibre Channel family of protocol standards. This family includes INCITS FC-FS-4, specifying the Framing and Signaling Interface; INCITS FC-LS-3, specifying Extended Link Services; INCITS FC-SW-6, specifying Fabric behavior; and INCITS FC-SP-2, specifying Fibre Channel security protocols.

FC-GS-8 provides Generic Services that may be utilized by any upper layer protocol that makes use of Fibre Channel as a transport. These upper layer protocols are usually called FC-4s. Some FC-4s include INCITS FCP-4, a transport for SCSI; and INCITS FC-SB-6, a transport for the Single-Byte protocol.

Contents **Page**

Figure **Page**

Table **Page**

American National Standard
for Information Technology —

Fibre Channel —
Generic Services - 8 (FC-GS-8)

1 Scope

This standard describes in detail the Services accessed by well-known addresses defined in ~~FC-FS-4~~

Generic Services described in this document are:

a) Directory Service;

b) Management Service; and

c) Event Service.

In addition, to the aforementioned Generic Services, the Common Transport (CT) protocol is described. The Common Transport Service provides a common FC-4 for use by Generic Services.

~~The following commands, parameter data, and features defined in previous versions of this standard are made obsolete by this standard:~~

~~Annex B: Discovery (Informative)~~
~~Annex C: Time Service (Informative)~~
~~Annex D: Performance Server (Informative)~~

2 Normative References

2.1 Overview

The following standards contain provisions that, through reference in the text, constitute provisions of this standard. At the time of publication, the editions indicated were 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 standards listed below.

For electronic copies of some standards, visit ANSI’s Electronic Standards Store (ESS) at www.ansi.org. For printed versions of all standards listed here, contact Global Engineering Documents, 15 Inverness Way East, Englewood, CO; 80112-5704, (800) 854-7179.

Additional availability contact information is provided below as needed.

2.2 Approved references

ISO/IEC 646:1991, *Information technology - ISO 7-bit coded character set for information interchange (third edition)*

IEEE 802.3-2005, *Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications*

INCITS 4-1986 (R2002), *Information Systems - Coded Character Sets - 7-Bit American National Standard Code for Information Interchange (7-Bit ASCII)*

INCITS 332:1999, *Fibre Channel - Arbitrated Loop (FC-AL-2)*

INCITS 332:1999/AM1-2003, *Fibre Channel-Arbitrated Loop–2 (FC-AL-2) Amendment 1*

INCITS 475:2011 *Fibre Channel - Inter-Fabric Routing (FC-IFR)*

INCITS 481:2012, *Fibre Channel Protocol for SCSI - 4 (FCP-4)*

INCITS 488-2016, *Fibre Channel - Framing and Signaling - 4 (FC-FS-4)*

INCITS 496:2012, *Fibre Channel - Security Protocols - 2 (FC-SP-2)*

INCITS 508:2014, *Storage Management - Host Bus Adapter Application Programming Interface (SM-HBA-2)*

INCITS 509:2014, *Fibre Channel - Backbone - 6 (FC-BB-6)*

INCITS 511:2016, *Fibre Channel - Switch Fabric - 6 (FC-SW-6)*

INCITS 513:2014, *SCSI Primary Commands - 4 (SPC-4)*

INCITS 515:2016, *SCSI Architecture Model - 5 (SAM-5)*

INCITS TR-20:1998, *Fibre Channel - Fabric Loop Attachment (FC-FLA)*

INCITS TR-48:2012, *Fibre Channel - Methodologies for Interconnects (FC-MI-3)*

2.3 References under development

At the time of publication, the following referenced standards were still under development. For information on the current status of the documents, or regarding availability, contact the relevant standards body or other organization as indicated.

INCITS 487-201x, *Fibre Channel - Link Services - 3 (FC-LS-3)*

INCITS 502-201x, *SCSI Primary Commands - 5 (SPC-5)*

INCITS 545-201x, *Fibre Channel - Framing and Signaling - 5 (FC-FS-5)*

2.4 Other references

IETF Request for Comments (RFCs) may be obtained directly from the IETF web site at http://www.ietf.org/rfc.html.

RFC 1738, *Uniform Resource Locators (URL)*, December 1994.

RFC 2396, *Uniform Resource Identifiers (URI),* August 1998.

RFC 2732, *Format for Literal IPv6 Addresses in URL's*, December 1999.

RFC 3986, *Uniform Resource Identifier (URI),* January 2005.

RFC 4044, *Fibre Channel Management MIB,* May 2005.

RFC 4122, *A Universally Unique IDentifier (UUID) URN Namespace*, July 2005.

RFC 4248, *The Telnet URI Scheme,* October 2005.

RFC 4266, *The Gopher URI Scheme*, November 2005.

3 Definitions and conventions

3.1 Overview

For FC-GS-8, the following definitions, conventions, abbreviations, acronyms, and symbols apply.

3.2 Definitions

3.2.1 Active Zone Set
Zone Set that is currently active

3.2.2 address identifier
address value used to identify source (S_ID) or destination (D_ID) of a frame

3.2.3 Area_ID
second level in a three-level addressing hierarchy (see FC-SW-6)

3.2.4 Client
entity that makes requests of a Server via the Common Transport

3.2.5 Common Transport (CT)
protocol that provides access to Services and their related Servers

3.2.6 Controlling Switch
FC Switch including a FC Switching Element for its Principal Domain and a FC Switching Element for the Virtual Domain of a Distributed Switch (see FC-SW-6)

3.2.7 Default Zone
Zone that contains all Nx_Ports that are not members of any Zone in the Active Zone Set (see FC-SW-6)

3.2.8 Domain_ID
highest or most significant hierarchical level in the three-level addressing hierarchy (see FC-SW-6)

3.2.9 Fabric (see FC-SW-6)

3.2.10 Fabric VE ID:
{N_Port_ID, Local VE ID} pair of values used to identify a VE within a Fabric

3.2.11 FC_Port (see FC-FS-4)

3.2.12 F/NL_Port (see FC-AL-2)

3.2.13 gateway (see FC-FS-4)

3.2.14 Generic Services
collection of Services defined by this standard

3.2.15 Global VE ID
identifier used to uniquely identify a Virtual Entity

Note 1 to entry: For example a 16-byte Universally Unique IDentifier (UUID, see RFC 4122)

3.2.16 Hard Zoning enforcement
Zoning technique in which the Fabric limits frame exchange using frame-by-frame filtering

3.2.17 Host (see FC-FS-4)

3.2.18 Host Bus Adapter
I/O adapter that connects a host computer bus to the Fibre Channel medium

3.2.19 hub (see FC-FS-4)

3.2.20 Interconnect Element
device in a Fabric that assists in the transport of Fibre Channel frames between VN_Ports

3.2.21 link (see FC-FS-4)

3.2.22 Local VE ID
value used to locally identify a VE within a VEM

Note 1 to entry: The local VE ID has a scope local to an N_Port_ID and is carried in the Priority field of the FC header.

3.2.23 Logical Port
port supporting mappings of a VN_Port, VF_Port, VE_Port, or VA_Port

3.2.24 Multi-function device (see FC-LS-3)

3.2.25 NAS server (see FC-LS-3)

3.2.26 N_Port (see FC-FS-4)

3.2.27 NL_Port (see FC-FS-4)

3.2.28 Nx_Port (see FC-FS-4)

3.2.29 N_Port_ID (see FC-FS-4)

3.2.30 N_Port_Name (see FC-FS-4)

3.2.31 node (see FC-FS-4)

3.2.32 Platform
container for one or more nodes and one or more LCFs

Note 1 to entry: A Platform may have functions beyond those of nodes and LCFs (see 6.2.3.4.3).

3.2.33 Physical Object
container for one or more Interconnect Elements

3.2.34 Physical Port
physical entity that supports a Logical Port

3.2.35 PN_Port (see FC-FS-4)

3.2.36 Port_Name
Name_Identifier associated with an FC_Port (see FC-FS-4)

3.2.37 printable ASCII characters
ASCII characters in the range 20h through 7Eh

3.2.38 Requesting_CT
CT that is sending a request

3.2.39 Responding_CT
CT that is responding to a prior request

3.2.40 Secured Association Identifier (SAID)
value that specifies a Secured Association relationship between two Clients

Note 1 to entry: SAIDs allow multiple concurrent Secured Associations to be active between two Clients.

3.2.41 Server
entity that, via its CT, accepts requests from a Client and provides responses to the Client

Note 1 to entry: A Server is accessed via a Service (e.g., the Name Server is accessed using the Directory Service).

3.2.42 Service
A Service is provided by a node or a Fabric, accessible via an N_Port that is addressed by a WKA or an N_Port_ID (e.g., the Directory Service and the Alias Service)

Note 1 to entry: A Service provides access to one or more Servers.

3.2.43 Server Session
series of one or more CT requests from the same Client to the same server that have been identified as related by being preceded by a Server Session Begin (SSB) CT request and followed by a Server Session End (SSE) CT request

3.2.44 Soft Zoning enforcement
Zoning technique in which the Fabric enforces membership through name server visibility

3.2.45 Switch (see FC-SW-6)

3.2.46 Symbolic Name
user-defined name for an object

Note 1 to entry: The Directory Service does not guarantee uniqueness of its value.

3.2.47 topology (see FC-SW-6)

3.2.48 Virtual Domain Supervisor
FC Switching Element representing the Virtual Domain of a Distributed Switch (see FC-SW-6)

3.2.49 VEM ID
identifier used to uniquely identify a VEM

Note 1 to entry: For example a 16-byte Universally Unique IDentifier (UUID, see RFC 4122).

3.2.50 Virtual Entities Manager
entity managing Virtual Entities

Note 1 to entry: For example a Hypervisor.

3.2.51 Virtual Entity (VE)
virtualized resource

Note 1 to entry: For example a Virtual Machine (VM).

3.2.52 Well-Known Address
address identifier defined in FC-FS-4 to access a Service

3.2.53 Zone (see FC-SW-6)

3.2.54 Zone Definition (see FC-SW-6)

3.2.55 Zone Member (see FC-SW-6)

3.2.56 Zone Member Definition (see FC-SW-6)

3.2.57 Zone Name (see FC-SW-6)

3.2.58 Zone Set (see FC-SW-6)

3.2.59 Zone Set Name (see FC-SW-6)

3.2.60 Zone Set State (see FC-SW-6)

3.3 Editorial Conventions

In FC-GS-8, a number of conditions, mechanisms, sequences, parameters, events, states, or similar terms are printed with the first letter of each word in uppercase and the rest lowercase (e.g., Exchange, Class). Any lowercase uses of these words have the normal technical English meanings.

Lists sequenced by lowercase or uppercase letters show no ordering relationship between the listed items.

EXAMPLE 1 - The following list shows no relationship between the colors:

a) red, specifically one of the following colors:

   A) crimson; or

   B) amber;

b) blue; or

c) green.

Lists sequenced by numbers show an ordering relationship between the listed items.

EXAMPLE 2 - The following list shows the order in which a page is meant to be read:

1) top;

2) middle; and

3) bottom.

The ISO/British convention of decimal number representation is used in this standard. Numbers may be separated by single spaces into groups of three digits counting from the decimal position, and a period is used as the decimal marker. A comparison of the ISO/British, ISO/French, and American conventions is shown in table 1.

Table 1 – ISO and American Conventions

ISO/British	ISO/French	American
0.6	0,6	0.6
3.14159265	3,141 592 65	3.14159265
1 000	1 000	1,000
1 323 462.9	1 323 462,9	1,323,462.9

In case of any conflict between figure, table, and text, the text, then tables, and finally figures take precedence. Exceptions to this convention are indicated in the appropriate sections.

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. Exceptions to this convention are indicated in the appropriate sections.

If a field or a control bit in a frame is specified as not meaningful, the entity that receives the frame shall not check that field or control bit.

If a field or a control bit in a frame is specified as reserved, the entity that sends the frame shall set the field or control bit to zero, and the entity that receives the frame shall not check that field or control bit.

When the value of the bit or field is not relevant, x or xx appears in place of a specific value. If a field or a control bit in a frame is specified as not meaningful, the entity that receives the frame shall not check that field or control bit.

Unless stated otherwise: numbers that are not immediately followed by lower-case b or h are decimal values; numbers immediately followed by lower-case b (xxb) are binary values; and numbers or upper case letters immediately followed by lower-case h (xxh) are hexadecimal values.

3.4 Abbreviations, acronyms and symbols

Abbreviations and acronyms applicable to this standard are listed. Definitions of several of these items are included in 3.2.

B_Port Bridge Port (see FC-SW-6)
CT Common Transport
CT_IU Common Transport Information Unit

D_ID	Destination address identifier
E_Port	Expansion Port (see FC-SW-6)
ELS	Extended Link Service
F_Port	Fabric Port (see FC-FS-4)
FL_Port	Fabric Loop Port (see FC-AL-2 and FC-SW-6)
Fx_Port	F_Port or FL_Port
GBIC	Gigabit Interface Converter
GE	Gigabit Ethernet
GFC	Gigabit Fibre Channel
GLM	Gigabit Link Module
GS	Fibre Channel Generic Service
HBA	Host Bus Adapter
IN_ID	Initial Identifier
IP	Internet Protocol
IU	Information Unit
LCF	Link Control Facility (see FC-FS-4)
LUID	Logical Unit Unique Identifier (see ~~FC-HBA~~)
LUN	Logical Unit Number (see ~~SAM-3~~)
NS	Name Server
NSM	Native SNMP Mapping
RNID	Request Node Identification Data Extended Link Service
SFP	Small Form-factor Pluggable
SI	Sequence Initiative
S_ID	Source address identifier
SNMP	Simple Network Management Protocol
TS	Time Service
TSAP	Time Service Access Protocol
UDP	User Datagram Protocol
ULP	Upper Level Protocols
VDS	Virtual Domain Supervisor
WKA	Well-Known Address

3.5 Keywords

3.5.1 expected: A keyword used to describe the behavior of the hardware or software in the design models assumed by this standard. Other hardware and software design models may also be implemented.

3.5.2 ignored: A keyword used to describe an unused bit, byte, word, field or code value. The contents or value of an ignored bit, byte, word, field or code value shall not be examined by the receiving device and may be set to any value by the transmitting device.

3.5.3 invalid: A keyword used to describe an illegal or unsupported bit, byte, word, field or code value. Receipt of an invalid bit, byte, word, field or code value shall be reported as an error.

3.5.4 mandatory: A keyword indicating an item that is required to be implemented as defined in this standard.

3.5.5 may: A keyword that indicates flexibility of choice with no implied preference (equivalent to “may or may not”).

3.5.6 may not: A keyword that indicates flexibility of choice with no implied preference (equivalent to “may or may not”).

3.5.7 obsolete: A keyword indicating that an item was defined in prior Fibre Channel standards but has been removed from this standard.

3.5.8 optional: A keyword that describes features that are not required to be implemented by this standard. However, if any optional feature defined by this standards is implemented, then it shall be implemented as defined in this standard.

3.5.9 reserved: A keyword referring to bits, bytes, words, fields and code values that are set aside for future standardization. A reserved bit, byte, word or field shall be set to zero, or in accordance with a future extension to this standard. Recipients should not check reserved bits, bytes, words or fields for zero values. Receipt of reserved code values in defined fields shall be reported as error.

3.5.10 restricted: A keyword referring to bits, bytes, words, and fields that are set aside for use in other Fibre Channel standards. A restricted bit, byte, word, or field shall be treated as a reserved bit, byte, word or field for the purposes of the requirements defined in this standard.

3.5.11 shall: A keyword indicating a mandatory requirement. Designers are required to implement all such mandatory requirements to ensure interoperability with other products that conform to this standard.

3.5.12 should: A keyword indicating flexibility of choice with a strongly preferred alternative; equivalent to the phrase “it is strongly recommended”.

3.5.13 x or **xx:** The value of the bit or field is not relevant.

3.6 T10 Vendor ID Fields

A T10 Vendor ID shall be a string of one to eight characters that is recorded in an informal list of Vendor IDs maintained by INCITS Technical Committee T10 (see http://www.t10.org, ~~SPC-3~~).

A field described as containing a T10 Vendor ID shall contain the first character of the T10 Vendor ID in the first byte of the field, and successive characters of the T10 Vendor ID in successive bytes of the field. Any bytes of the field not filled by characters of the T10 Vendor ID shall be filled with ASCII space characters (20h).

4 Common Transport for Generic Services (CT)

4.1 Overview

Fibre Channel Generic Services share a Common Transport (CT) at the FC-4 level. The CT provides access to a Service (e.g. Directory Service) with a set of Service parameters that facilitates the usage of Fibre Channel constructs. It also provides another level of multiplexing that simplifies the Server-to-Server communication for a distributed Service. It is important to note that Fibre Channel Generic Services do not require a high performance communication channel as do high performance I/O protocols (e.g., SCSI, IP, VI). The relationship of CT with respect to Generic Services and FC-FS is illustrated in figure 1.

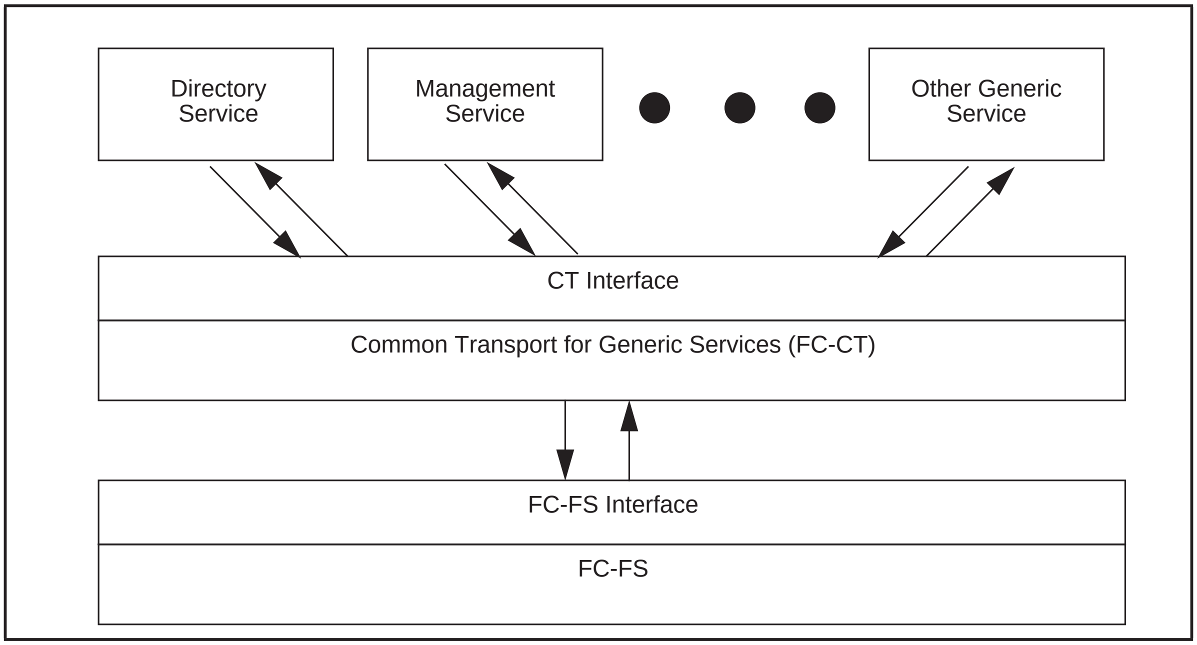


Figure 1 – Relationship of the Common Transport with Generic Services and FC-FS

The CT provides access to a Service that may then provide access to more than one Server. Each Server may provide access to different information or controls, or each Server may provide a different access model to the same information and controls. The Service is identified by its GS_Type and accessed at a Well-Known Address specific to the Service or at any N_Port_ID where it is offered. A Well-Known Address shall not be subject to Zone restrictions (i.e., a Well-Known Address is always accessible, irrespective of the current Active Zone Set). The Server beneath the Service is referenced by its GS_Subtype within the GS_Type. The relationship between CT, a Service, and its Servers is illustrated in figure 2.

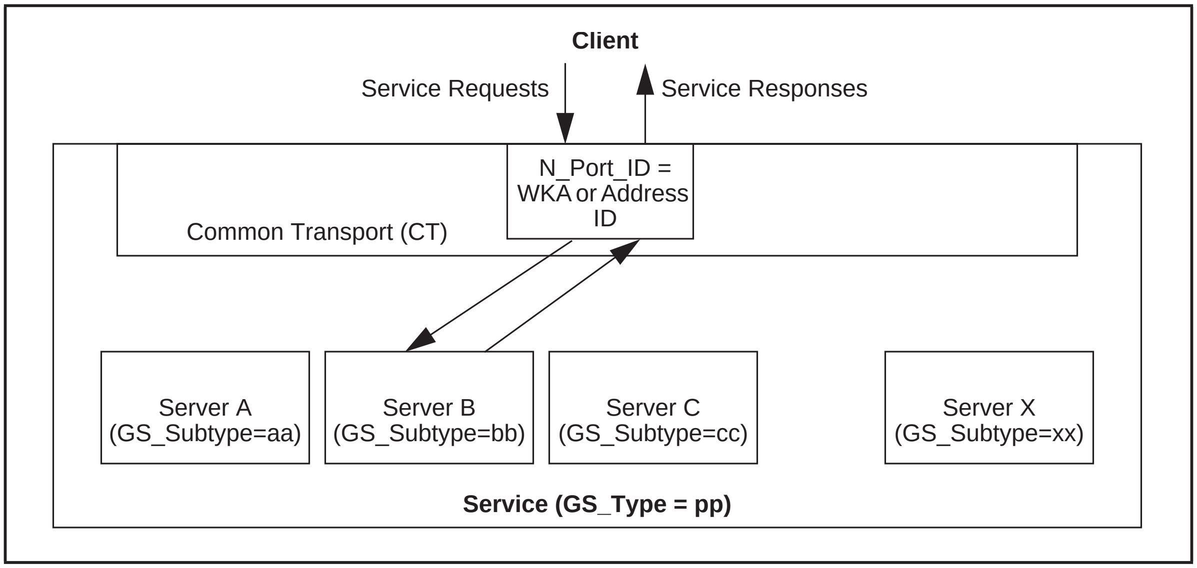


Figure 2 – Relationship between Common Transport, Service, and Servers

4.2 General concepts

The following parameters describe the information that the CT delivers to a Service:

a) type of Service and Server;

b) type of transaction;

c) class of service; and

d) maximum size of an IU.

The types of Services described by this document that are accessible via the CT are:

a) Directory Service;

b) Management Service; and

c) Event Service.

There are two types of transactions:

a) Request: An IU transmitted from one entity (i.e., a Client) to another entity (i.e., a Server) to request the Server to perform a function; and

b) Response: An IU transmitted from a Server to a Client responding to an earlier Request from the Client for the Server to perform a function.

The class of service is an indication of the quality of service that a Client expects from the underlying transport. The following classes of service may be used to send a CT_IU:

a) Class 2; or

b) Class 3.

Class 3 service, if available in the operational environment, shall also be available to the Client.

NOTE 1 – When the Server is distributed amongst several entities within the Fabric, Class F service may be used for communicating CT information between those entities. This usage of Class F is not defined by this standard (see FC-SW-6).

A Client may restrict the size of IUs it receives from a Server. A Server shall observe and obey this restriction on behalf of the Client. It may do so as described in 4.3.2.8.

4.3 CT information unit

4.3.1 Overview

A Common Transport IU (CT_IU) is the common Fibre Channel Sequence used to transfer all information between a Client and a Server. The first part of the CT_IU contains a preamble with information common to all CT_IUs. The CT_IU preamble shall consist of the 16-byte basic CT_IU preamble. Additionally, one or more of the following optional preambles may be used:

a) Extended CT_IU preamble; or

b) Vendor Specific preamble.

If more than a single preamble is used with in a CT_IU, the preamble order shall be as shown in table 2. All preambles present in the CT_IU (e.g., basic CT_IU preamble, extended CT_IU preamble, and Vendor Specific preamble) shall be defined as the CT_IU preamble. The remainder of the CT_IU contains additional information as defined by the preamble, and may be zero or more bytes in length. The format of the CT_IU is shown in table 2.

Table 2 – CT_IU

Word **Bits**	**3322 2222** **1098 7654**	**2222 1111** **3210 9876**	**1111 1100** **5432 1098**	**0000 0000** **7654 3210**
0 to 3	basic CT_IU preamble			
4 to 25	extended CT_IU preamble (if required)			
n to n+34	Vendor Specific preamble (if present)			
	additional information			

CT_IU preamble

4.3.2 Basic CT_IU preamble description

4.3.2.1 Overview

The 16-bytes of the basic CT_IU preamble are defined as follows. The format is shown in table 3.

Table 3 – Basic CT_IU preamble

Word Bits	**3322 2222 1098 7654**	**2222 1111 3210 9876**	**1111 1100 5432 1098**	**0000 0000 7654 3210**
0	Revision	IN_ID		
1	GS_Type	GS_Subtype	Options	Reserved
2	Command/Response code		Maximum/Residual Size	
3	Fragment ID	Reason Code	Reason Code Explanation	Vendor Specific

4.3.2.2 Revision Field

The Revision Field denotes the revision of the protocol. A version of 01h, or 02h, indicate prior versions of this standard. A value of 03h shall be used to indicate this version.

If a Responding_CT receives a Request CT_IU containing a Revision Field value that is higher than its supported value, the Responding_CT shall respond with its highest supported Revision Field value. The Requesting_CT is responsible for detecting and handling any incompatibility issues that may occur.

If a Responding_CT receives a Request CT_IU containing a Revision Field value that is equal to or lower than its supported value, the Responding_CT shall respond with the Revision Field value received in the Request CT_IU.

NOTE 2 – The version was changed to 03h to allow implementations to indicate support of new FC-GS-4 features (e.g. the Vendor Specific Preamble (see 4.3.4) and the common CT Requests (see 4.9).

4.3.2.3 IN_ID Field

This field shall be set to zero by the Requesting_CT.

NOTE 3 – The IN_ID field is provided to allow distributed Servers to communicate the identity of the original requestor. This field is not intended to enable third-party responses by distributed Servers.

4.3.2.4 GS_Type Field

This field is used to indicate the type of Service. The values are defined in table 4.

Table 4 – GS_Type values

Encoded value	Description
00h to 1Fh	Vendor Specific
20h	Reserved for use by FC-SW-5
F4h	Event Service
F7h	Reserved for Key Distribution Service - Obsolete
F8h	Reserved for Alias Service - Obsolete
FAh	Management Service
FBh	Reserved for Time Service - Obsolete
FCh	Directory Service
All Others	Reserved

4.3.2.5 GS_Subtype Field

This field indicates the specific Server behind the Service. Values in this field are provided by the individual Service.

NOTE 4 – The GS_Subtype field is used to indicate second level routing behind the N_ Port (e.g., if more than one Server is provided by the Directory Service at the WKA FFFFFCh, then the GS_Subtype field is used to distinguish these different Servers).

4.3.2.6 Options Field

This field denotes options used by the Requesting_CT or Responding_CT, as shown in table 5.

Table 5 – Options field bits

Bit Position	Description
7	Obsolete. Shall be set to zero for compliance with this standard.
6	Partial Response. A value of one indicates that the response is known to be incomplete. A value of zero indicates that the completeness of the response is not specified (i.e., when a Server is distributed amongst several entities (e.g., Switches), if one or more of the entities fails to respond, the Partial Response bit may be used to indicate that those entities did not participate in the answer given. Note that a zero value for this bit only indicates that the Server does not have any knowledge that the response is incomplete, or that it is unable to report that knowledge).
5 to 3	Encoded values - see table 6
2	Vendor Specific preamble. When the bit is set to one, the Vendor Specific preamble is present. When the bit is set to zero, the Vendor Specific preamble is not present (see 4.3.4).
1 to 0	Reserved

Table 6 – Options field bits 5 to 3 values

Encoded Values (binary)	Description
000	Extended preamble not present.
001 to 011	Reserved
100	Extended preamble present and all fields valid. Service may retain residual information (see 4.3.2.8).
101	Extended preamble present and all fields valid. Service shall retain residual information (see 4.3.2.8) for later retrieval using a Get More Information Request.
110	Extended preamble present, only transaction_ID field is valid. Service may retain residual information (see 4.3.2.8).
111	Extended preamble present, only transaction_ID field valid. Service shall retain residual information (see 4.3.2.8) for later retrieval using a Get More Information Request.

4.3.2.7 Command/Response code field

This field indicates whether the CT_IU is a request or a response. If the CT_IU is a request, this field then specifies the command to be performed. If the CT_IU is a response, then this field indicates whether the request was accepted or rejected. Requests and responses are further defined in 4.4.

Common Request CT_IUs have the same definition for all servers. Table 7 depicts the valid Command/Response code values.

Table 7 – Command/Response codes

Encoded Value	Description
0001h to 03FFh	Request CT_IU
0400h to 05FFh	Reserved for FC-SW-5
0600h to 7EFFh	Request CT_IU
7F00h to 7FFFh	Common Request CT_IU
8001h	Reject Response CT_IU
8002h	Accept Response CT_IU
E000h to FFFFh	Reserved for FC-SW-5
other values	Reserved

4.3.2.8 Maximum/Residual Size field

This field manages the size of the information returned in an Accept CT_IU. A Requesting_CT may specify the maximum size of the Accept CT_IU it is able to receive. If the Responding_CT has more available information to send than allowed by the maximum size, it shall indicate the excess residual size in the Accept CT_IU. The values for Maximum/Residual Size are interpreted as follows:

a) In the Request CT_IU:

   A) 0000h: No Maximum Size indicated; the Accept CT_IU may be any size;

   B) FFFFh: This value is reserved for the Request CT_IU; or

   C) Any other value: The Encoded Value indicates the maximum number of words that shall be sent in the Accept CT_IU, not inclusive of the CT_IU preamble.

b) In the Accept CT_IU:

   A) 0000h: All available information was returned in the Accept CT_IU;

   B) FFFFh: The Encoded Value indicates greater than 65 534 available words of information were not sent in the Accept CT_IU, in excess of the Maximum Size specified in the Request CT_IU; or

   C) Any other value: The Encoded Value indicates the number of available words of information that were not sent in the Accept CT_IU, in excess of the Maximum Size specified in the Request CT_IU.

   or

c) For the Reject CT_IU and any other CT_IU, this field is reserved.

4.3.2.9 Fragment ID field

The fragment ID field contains a value that identifies the fragment contained in the IU. The value contained in this field in the Request CT_IU shall be echoed by the Service in the associated Response CT_IU.

4.3.2.10 Reason Code field

The Reason Code field contains the reason code associated with a Reject CT_IU (see 4.4.4). When the Command/Response code field indicates the CT_IU is not a Reject CT_IU, this field is reserved.

4.3.2.11 Reason Code Explanation field

This field contains a reason code explanation associated with a Reject CT_IU (see 4.4.4). When the Command/Response code field indicates the CT_IU is not a Reject CT_IU, this field is reserved.

4.3.2.12 Vendor specific field

This field contains a vendor specific reason code associated with a Reject CT_IU (see 4.4.4). When the Command/Response code field indicates the CT_IU is not a Reject CT_IU, this field is reserved.

4.3.3 Extended CT_IU preamble description

4.3.3.1 Overview

The 88-bytes of the extended CT_IU preamble are defined as follows. The format is shown in table 8. The extended CT_IU preamble is preceeded by the 16-bytes of the basic CT_IU preamble (see 4.3.2). The calculation of the Authentication Hash Block is defined in 4.8.

Table 8 – Extended CT_IU preamble

Word Bits	3322 2222 1098 7654	2222 1111 3210 9876	1111 1100 5432 1098	0000 0000 7654 3210
4	Authentication SAID			
5	transaction_id			
6 to 7	Requesting_CT N_Port_Name			
8 to 9	Time Stamp			
10 to 25	Authentication Hash Block			

4.3.3.2 Authentication SAID

This field denotes the Secured Association Identifier that uniquely identifies the algorithm and key used to generate the Authentication Hash Block, as pre-arranged between the Requesting_CT and the Responding_CT. The SAID Value shall be the value specified in FC-SP-2 when using the FC-SP-

2 SA Management protocol. If the FC-SP-2 SA Management protocol is not used the value is outside the scope of this standard.

4.3.3.3 transaction_id

The transaction_id field contains an opaque value (e.g., managed and controlled by the Requesting_CT). The opaque value shall not be validated by the Responding_CT. The Responding_CT shall respond using the same encoded transaction_id value in the Accept CT_IU or Reject CT_IU as was supplied in the Request CT_IU.

4.3.3.4 Requesting_CT N_Port_Name

This field contains the value of the N_Port_Name of the Requesting_CT. The Responding_CT shall respond using the same encoded N_Port_Name value in the Accept CT_IU or Reject CT_IU as was supplied in the Request CT_IU.

4.3.3.5 Time Stamp

This field contains a time stamp value set by the CT sending the CT_IU. The Requesting_CT shall set this value according to its time reference when it sends the CT_IU. The Responding_CT may set this value according to its time reference when it sends the CT_IU, or it may echo the value sent by the Requesting_CT. In all cases, the value of the time stamp shall be consistently increasing (i.e., the value shall not decrease or be random in subsequent requests or responses).

4.3.3.6 Authentication Hash Block

This field contains the encoded value of the hash generated by the identified algorithm and key (see 4.3.3.2).

4.3.4 Vendor Specific CT_IU Preamble

4.3.4.1 Overview

The Options field of the Basic FC-CT preamble indicates whether the Vendor Specific preamble is present (see 4.3.2.6). If the Vendor Specific preamble is present, it shall immediately follow the Basic CT_IU preamble if no Extended CT_IU preamble exists, or it shall immediately follow the Extended CT_IU preamble if that preamble does exist.

The format of the Vendor Specific CT_IU preamble is shown in table 9.

Table 9 – Vendor Specific CT_IU preamble

Word Bits	3322 2222 1098 7654	2222 1111 3210 9876	1111 1100 5432 1098	0000 0000 7654 3210
0 to 1	Vendor Identifier			
2 to 33	Vendor Specific Information			

4.3.4.2 Vendor Identifier

Contains the T10 Vendor ID (see 3.6) of the vendor that defines the content of the Vendor Specific Information field.

4.3.4.3 Vendor Specific Information

This field contains 32 words of vendor specific information.

4.3.4.4 Vendor Specific Reject Information

4.3.4.4.1 General Description

When the Vendor Specific preamble is present, additional Vendor Specific reject information may be provided as part of the Vendor Specific Information. The vendor shall be responsible for specifying any appropriate Vendor Specific reject information as part of their Vendor Specific Information structures. Any Vendor Specific reject information is provided in addition to the reason code information and reason code explanation information specified for each CT command.

4.3.4.4.2 Reject CT_IU Reason Code Field

This field shall indicate the appropriate reject CT_IU reason code values as defined for each CT command. If the CT command is directed to a Vendor Specific server as defined by the GS_Type or the GS_Subtype fields, then the Vendor Specific reject information may be provided without specifying a value in the Reason Code field.

4.3.4.4.3 Reject CT_IU Reason Code Explanation Field

This field shall indicate the appropriate reject CT_IU reason code explanation values as defined for each CT command. If the CT command is directed to a Vendor Specific server as defined by the GS_Type or the GS_Subtype fields, then the Vendor Specific reject information may be provided without specifying a value in the Reason Code Explanation field.

4.3.4.4.4 Reject CT_IU Vendor Specific Field

When Vendor Specific reject information is returned, the Reject CT_IU Vendor Specific field shall be set to FFh which indicates that additional Vendor Specific reject information is contained in the Vendor Specific preamble.

4.3.4.5 Processing of the Vendor Specific Preamble

The processing of the Vendor Specific preamble shall be subject to the following rules:

a) If the information contained in the Vendor Specific preamble is not recognized or processed by the server, then the command proceeds as defined and the server shall not return the Vendor Specific preamble in the response (i.e., Accept or Reject); and

b) For any CT command defined in this standard, the Vendor Specific information shall not cause the server to exhibit any behavior, or operate in a manner different from that defined in the standard, and each CT command shall conform with all the requirements stated in this standard.

4.3.5 CT_IU Additional information

Following the CT_IU preamble and optional extended preamble, the additional information bytes contain information specific to the GS_Type, GS_Subtype, and Command/Response code.

4.4 CT Information Units (CT_IU)

4.4.1 Overview

A set of Server request and response CT_IUs are defined by CT for use by the Generic Services. One Request CT_IU and two Response CT_IUs - Accept CT_IU and Reject CT_IU - are defined.

4.4.2 Request CT_IU

A Request CT_IU is a CT_IU in which the Command/Response code field contains a command code value.

The command code shall define the particular request that is to be executed by the Server. The range of codes for Common Request CT_IUs are defined in 4.9, and shall have identical definitions for all servers. The other command codes shall be defined independently by each Server.

The Server shall verify that the overall length of each Request CT_IU is valid for the command operation code contained in the CT_IU. If an invalid CT_IU size is detected, a Reject CT_IU (see 4.4.4) shall be returned with the reason code set to ‘Invalid CT_IU size’ and the reason code explanation set to ‘No additional explanation’.

The additional information in the CT_IU contains the associated command specific data. The associated command specific data shall be defined independently by each Server, based on the command code.

4.4.3 Accept CT_IU

An Accept CT_IU is a CT_IU in which the Command/Response code field contains a value of 8002h. The Accept CT_IU shall notify the Client that the request has been successfully completed.

The additional information in the CT_IU contains the associated response specific data. The associated response specific data shall be defined independently by each Server.

4.4.4 Reject CT_IU

A Reject CT_IU is a CT_IU in which the Command/Response code field contains a value of 8001h. The Reject CT_IU shall notify the Client that the request has been unsuccessfully completed.

The reason code indicates the general reason why the request was rejected. Table 10 indicates the defined Reject CT_IU Reason Codes.

The vendor specific field may be used by Vendors to specify additional reason codes.

Table 10 – Reject CT_IU Reason Codes

Value	Description
01h	Invalid command code
02h	Invalid version level
03h	Logical error
04h	Invalid CT_IU size
05h	Logical busy
07h	Protocol error
09h	Unable to perform command request
0Bh	Command not supported
0Dh	Server not available
0Eh	Session could not be established
others	Reserved
FFh	Vendor specific error (see Vendor Specific field)

Invalid command code: The command code passed in the Request CT_IU is not defined by the Server.

Invalid version level: The specified version level is not supported by the Server.

Logical error: The request identified by the Request CT_IU command code and additional information content is invalid or logically inconsistent for the conditions present.

Invalid CT_IU size: The CT_IU size is invalid for the Request CT_IU command code.

Logical busy: The Server is logically busy and unable to process the request at this time.

Protocol error: This indicates that an error has been detected that violates the rules of the Server protocol that are not specified by other error codes.

Unable to perform command request: The Server is unable to perform the request.

Command not supported: The Server does not support the command requested.

Server not available: The server identified by the GS_Type and GS_Subtype is not available.

Session could not be established: A Server Session (i.e., using SSB) could not be established.

Vendor specific error: The Vendor Specific Field may be used by Vendors to specify additional reason codes.

The reason code explanation further defines the indicated reason code. If a request is rejected with a reason code of ‘Unable to perform command request’, then one of the reason code explanations, shown in table 11, may be returned. Additional per Server reason code explanations may also be returned.

Table 11 – Reject CT_IU Reason Code explanations for all Servers

Value	Description
00h	No additional explanation
F0h	Authorization exception
F1h	Authentication exception
F2h	Data base full
F3h	Data base empty
F4h	Processing request
F5h	Unable to verify connection
F6h	Devices not in a common Zone
F7h	Session established by another Client
Others	Reserved[a]

[a] Servers defined in this standard may also define reason code explanation values.

The use of these codes is further defined as follows:

a) If a request is rejected by the Server because the requestor is not authorized to make that request, then the Reject CT_IU reason code shall be ‘Unable to perform command request’, with a reason code explanation of ‘Authorization exception’; or

b) If a request is rejected by the Server because the requestor fails authentication, then the Reject CT_IU reason code shall be ‘Unable to perform command request’, with a reason code explanation of ‘Authentication exception’.

Additional reason code explanations may be defined by each Server. In the case of a conflict between an explanation code defined by a specific server and the codes defined in table 11, the specific server definition shall be used.

4.5 FC-2 mapping

4.5.1 Overview

The CT maps CT_IUs into the appropriate Fibre Channel constructs.

4.5.2 Fabric Login and N_Port Login

An Nx_Port shall perform Fabric Login, and shall perform N_Port Login with the WKA or N_Port_ID where the Service is offered, in the manner that is specified in FC-LS-3, before making any requests of a Server provided by the Service. An Nx_Port that has completed its registration with a Server shall perform explicit N_Port Logout with the Service. An Nx_Port that has completed any other requests with a Server should also perform explicit N_Port Logout with the Service.

A Service may perform N_Port Logout (LOGO) if it becomes resource constrained. A Service may use a least recently used algorithm in determining which entity to Logout.

4.5.3 Class of Service

Any Class of Service permitted in 4.2 may be used for the Request CT_IU.

The Class of Service of the Response CT_IU shall be the same as the Class of Service used for the corresponding Request CT_IU.

4.5.4 Exchange and Sequence management

Each Request CT_IU shall be the first Sequence of a new Exchange, and its associated Response CT_IU shall be the last Sequence of the same Exchange.

The SI is transferred at the end of a CT_IU transmission. If the Responding_CT does not have the SI at the end of a CT_IU reception, it shall consider this to be a protocol error and shall terminate the Exchange (see 4.5.7).

An initiator of a CT request sequence to or from an Nx_Port in Class 3 shall consider the request Sequence open until:

a) the initiator of the CT request sequence receives a frame of a CT reply sequence in the same Exchange; or

b) other rules for Sequence management in FC-FS-4 allow the request Sequence to be considered no longer open to its initiator.

Each CT_IU shall be mapped into a Sequence as shown in table 12.

Table 12 – CT_IU mapping to Sequences

CT_IU Name	Information Category	Payload	F/M/L[a]	SI[b]	M/O[c]
Request	2	One Request CT_IU	F	T	M
Response	3	One Response CT_IU	L	T	M

[a] The F/M/L column indicates whether the Sequence may be the First, Middle, or Last Sequence of the Exchange.
[b] The SI column indicates whether Sequence Initiative is Held or Transferred.
[c] The M/O column indicates whether support for the Sequence is Mandatory or Optional.

4.5.5 FC-2 interface

4.5.5.1 Overview

The mapping of CT_IUs to Fibre Channel Sequences and frames is described.

4.5.5.2 Routing bits

The routing bits shall be set to FC-4 device data (0000b).

4.5.5.3 Information category

This parameter shall be set as indicated in table 12.

4.5.5.4 D_ID

The D_ID shall identify the destination Fibre Channel address identifier for the CT_IU. For Request CT_IUs, this is the WKA or N_Port_ID where the Service is offered. For Response CT_IUs, this is the address identifier of the requesting Nx_Port.

4.5.5.5 S_ID

The S_ID shall identify the source Fibre Channel address identifier for the CT_IU. For Request CT_IUs, this is the address identifier of the requesting Nx_Port. For Response CT_IUs, this is the WKA or N_Port_ID where the Service is offered.

4.5.5.6 Type

The CT shall set this parameter to Common Transport (20h).

4.5.5.7 First Sequence

The Requesting_CT shall set this parameter true to originate a new Exchange in order to transmit a Request CT_IU.

4.5.5.8 Last Sequence

The Service shall set this parameter true in a Response CT_IU to terminate an Exchange at the end of the transaction.

4.5.5.9 Sequence Initiative

This parameter shall be set as described in 4.5.4.

4.5.5.10 Exchange reassembly

The CT shall set this parameter to zero.

4.5.5.11 Relative offset

Relative offset shall be used. Each CT_IU shall be treated as a continuous data block ~~by the FC-FS~~ and the initial relative offset of each CT_IU shall be set to zero.

4.5.5.12 Optional headers

The use of any Optional Header is not defined by this standard.

4.5.6 Correlation of requests and responses

The correlation of Request CT_IUs and Response CT_IUs shall be managed by the specific Service. FC-CT provides no generic mechanism for this management.

NOTE 5 – Services typically correlate a request to a response through the use of Exchange IDs.

4.5.7 Error detection and recovery

There are two levels of error recovery that may be effected:

a) invalid CT_IU or FC-CT protocol error;

   A) invalid/undefined GS_Type;

   B) invalid revision level;

   C) invalid Options; or

   D) Sequence payload exceeds the maximum size of CT_IU at a destination FC_CT.

   and

b) FC-FS protocol errors and timeouts;

   A) Sequence errors;

   B) request to response timeout; or

   C) Exchange errors.

When an invalid CT_IU or FC-CT protocol error is recognized, the Responding_CT indicates the error condition to the Requesting_CT using a Reject CT_IU.

When an FC-FS-4 protocol error is detected, the Exchange shall be terminated. Transactions shall be terminated by the Exchange Originator or the Exchange Responder using the Abort Sequence Link Service with the ~~LS_bit~~ set to terminate the Exchange.

All error conditions shall also be indicated to the Client.

4.6 Time constants

4.6.1 Overview

The following timeout values are defined for CT operations.

4.6.2 Request to response time

If the Requesting_CT does not receive a Response CT_IU from the Responding_CT within three times R_A_TOV, it shall consider this to be a protocol error.

4.6.3 Database propagation delay

A time lag may exist between successful completion of a request that causes an update to a database, and the time that the updated information is returned in response to a query request. This time lag is implementation and system dependent but shall not exceed six times R_A_TOV.

NOTE 6 – For example, consider a large Fabric with distributed Servers in each Switch. A registration request is completed at one point in the Fabric. At a point distant from the first point of the Fabric, a query is made related to the just-registered information. The query may return “stale” information because the new information has not yet been distributed across the Fabric. Or, the local distributed Server responding to the query may not yet have received an indication that its local cache of information needs to be refreshed.

4.7 Persistence of actions after Logout

In order to conserve resources, a Client should Logout with a Server once it has completed its activity with a Server. Similarly, a Server may Logout with a Client with which it has no open Exchanges.

In either case, any actions taken by the Server on its data in response to prior Client commands shall persist following the Logout and following any subsequent Login, except that the persistence of actions within a Server Session (see 4.9.5) shall be defined by each server.

Exceptions to this rule may be defined by a specific Server.

NOTE 7 – The idea is to allow a Client to Login with a Server, to then register information with that Server, and to Logout with that Server, and have the registered information remain registered at the Server. This allows a device to, for example, register itself with the Name Server and not have to then maintain the Login to remain registered.

4.8 CT_Authentication

4.8.1 Overview

CT_Authentication may be used to authenticate and optionally to encrypt CT_IU Requests and Responses between a Client and a Server. CT_Authentication is provided through the use of the extended CT_IU preamble (see 4.3.3). This mechanism provides a means to validate that requests and responses are transferred without change between authorized CTs, and distinguish these from requests and responses that are corrupted by an intermediary agent or generated by a non-authorized CT (i.e., integrity and anti-replay protection). Optionally this mechanism provides also confidentiality to CT_IUs (see 4.8.2). Use of CT_Authentication by specific Services and commands within those Services is not defined by this standard.

NOTE 8 – Use of CT_Authentication or other methods to ensure message integrity and authentication (see FC-SP-2) is recommended for use within the Management Service.

If CT_Authentication is invoked in a Request CT_IU, the corresponding Reject CT_IU or Accept CT_IU shall also use CT_Authentication.

For the following cases, either a Reject CT_IU without the extended preamble or a Reject CT_IU with an extended preamble indicating only the transaction_ID field is valid (see table 6, encoded values 110b and 111b) may be sent as a response:

a) Authentication failed; or

b) Failure that generates a Reject CT_IU occurred prior to authentication.

To provide CT_IU integrity and anti-reply protection, the CT_Authentication mechanism computes, using a specific algorithm and secret key, a hash that is a message authentication code for the associated CT_IU. In order to transfer an authenticated CT_IU, the hash is transferred as part of the CT_IU in the extended CT_IU preamble. Upon receiving the CT_IU, the receiver computes the hash using the same algorithm and secret key.

The received CT_IU is determined to be authentic if the received hash is identical to the value computed by the receiver, if the time stamp contained in the extended CT_IU preamble is greater than the previous time stamp received within the same Secured Association, and the time stamp is within acceptable limits.

NOTE 9 – The acceptable limits on time stamp values are not defined by this standard. The limits should be established by an external policy.

The received CT_IU is discarded and rejected if it is a Request CT_IU if the above criteria are not met. The receiver may apply additional CT_Authentication criteria to the CT_IU that are not defined by this standard.

Optionally, CT_Authentication may provide also confidentiality protection to CT_IUs (see 4.8.2).

Management and distribution of a common time reference may be accomplished through use of the Time Server, or by other means not defined by this standard. A Security Association needs to be established in order to use the CT_Authentication mechanism. One way to establish Security Associations is defined in FC-SP-2.

The hash is communicated in the Authentication Hash Block field of the extended CT_IU preamble. The value of the hash shall be computed over all data words of the CT_IU, with the exception of the Frame_Header, the IN_ID field in the basic CT_IU preamble, the Authentication Hash Block itself, and the frame CRC field. See FC-SP-2 for the definition of algorithms and parameters to be used to compute the hash.

4.8.2 CT Confidentiality

The CT Confidentiality transform is a cryptographic transform of the CT_IU payload, used in combination with the extended CT_IU preamble. The CT Confidentiality Encapsulation, shown in table 13, provides confidentiality to CT_IUs, in addition to the integrity and anti-replay protection provided by the extended CT_IU preamble.

Table 13 – CT Confidentiality Encapsulation

<table>
<tr><th>Word
Bits</th><th>3322 2222
1098 7654</th><th>2222 1111
3210 9876</th><th>1111 1100
5432 1098</th><th>0000 0000
7654 3210</th><th></th></tr>
<tr><td>0 to 3</td><td colspan="4">Basic CT_IU preamble</td><td rowspan="3">CT_IU
preamble</td></tr>
<tr><td>4 to 25</td><td colspan="4">Extended CT_IU preamble</td></tr>
<tr><td>26 to 60</td><td colspan="4">Vendor Specific CT_IU preamble
(if present)</td></tr>
<tr><td>N to P</td><td colspan="4">Payload Data
(additional information)</td><td rowspan="3">CT_IU
payload</td></tr>
<tr><td rowspan="2">P+1 to Q</td><td colspan="4">Padding</td></tr>
<tr><td colspan="2"></td><td>Pad Length</td><td>Not meaningful</td></tr>
</table>

A CT Confidentiality encapsulated CT_IU consists of a basic CT_IU preamble, an extended CT_IU preamble, an optional Vendor Specific preamble, the Payload Data, and the Padding words. The Payload Data and the Padding words constitutes the CT_IU payload of a CT Confidentiality encapsulated CT_IU, and are transformed according to the cryptographic Transforms applied to protect the confidentiality of the CT_IU payload.

The CT Confidentiality encapsulation of the payload data of a CT_IU is equivalent to the ESP_Header encapsulation of the Data Field of an FC-2 frame, as defined by FC-FS-4. The padding words are in variable numbers, according to the specific encryption algorithm used. Computation of the padding shall be as specified by FC-FS-4 for the ESP_Header.

Two cryptographic transforms are part of a CT Confidentiality encapsulation: one integrity algorithm, and one encryption algorithm. In transmission, the encryption transform is applied first, and the integrity transform is computed over the encrypted Information Unit. On receiving, the integrity of the CT_IU is checked first, then the encrypted CT_IU is decrypted.

See FC-SP-2 for definition of encryption algorithms and Transforms for CT_Authentication. The parameters and state used by the CT entities to select the usage of CT Confidentiality are negotiated via the FC-SP-2 SA management protocol.

4.9 Common Requests

4.9.1 Description

Common Requests may be executed for all server types.

4.9.2 CT_IU preamble values

The following values shall be set in the CT_IU preamble for these requests and their responses; fields not specified here shall be set as defined in 4.3.2:

a) GS_Type and GS_Subtype shall be set as defined by the server to which the common request is applied; and

b) Command Code: see table 14 for Request command codes.

Table 14 – Common Request Command Codes

Code	Mnemonic	Description	Reference subclause
7FF8h	GMI	Get More Information	4.9.4
7FF9h	SSB	Server Session Begin	4.9.5.2
7FFAh	SSE	Server Session End	4.9.5.3
7FFBh	ASYNC	Asynchronous Notification	4.9.6

4.9.3 Reason code explanations

A Reject CT_IU (see 4.4.4) shall notify the requestor that the request has been unsuccessful. The first error condition encountered shall be the error reported by the Reject CT_IU.

If a valid Common request is not received, the request is rejected with a reason code of “Invalid command code” and a reason code explanation of ‘No additional explanation’.

If a Common request is rejected with a reason code of ‘Unable to perform command request’, then one of the reason code explanations that are common to all servers, shown in table 11, is returned.

4.9.4 Get More Information (GMI)

The GMI shall be used by a requestor to obtain additional information for a previous request for which it received an Accept CT_IU with a value in the Residual Size field not equal to 0000h (see 4.3.2.8).

After a period of 6 times R_A_TOV has elapsed from receipt of the original Request CT_IU, or from a subsequent GMI CT_IU, a Service may discard any remaining information associated with the transaction.

The format of the GMI Request CT_IU is shown in table 15. The requestor supplies an original Transaction ID that shall be the value that was contained in the Transaction ID field of the Extended Preamble of the original Request CT_IU. An Information Offset is also supplied that is an offset into the entire information set created by the Service in response to the original Request CT_IU. The Information Offset shall be expressed in a number of words, with 0000h representing the first word of the set. The Fragment ID shall contain a value to identify the Fragment being requested, and the value shall be echoed in the Basic Preamble of the Accept CT_IU generated by the Server in response to the GMI Request CT_IU (see 4.3.2.9).

Table 15 – GMI Request CT_IU

Item	Size (Bytes)
CT_IU Preamble	see 4.3
original transaction id	4
Information Offset	4
reserved	3
Fragment ID	1

The Accept CT_IU contains the information forming the requested fragment.

The results are not specified if a GMI request is sent before the response to the request it references arrives.

4.9.5 Server Registration and State Change Notification Initiation

4.9.5.1 Delineating Server Sessions

A Server Session is defined to represent a collection of one or more requests for that server. In order for a Client to delineate a session, a Client notifies that particular server that a collection of requests is beginning and when the collection is complete. Sessions are initiated for a particular WKA such as the name service.

This standard defines two methods to signal the start and end of a session for a Service. The Server Session Begin and Server Session End methods are CT-Based requests that provide the recipient with information describing the GS_type and subtype for the session is being bracketed.

A session shall be terminated by Logout of the Client from the Server or Logout of the Server from the Client; however, the results of terminating a session by Logout may be defined by each server and may be different from the results of terminating a session by an SSE command.

Other interfaces (e.g., SNMP or SMI-S) may support Server Sessions. When one of those interfaces indicates the beginning of a Server Session, and has not indicated the end of the Server Session, then this shall be reported to all CT Clients as if a Server Session Begin request had been received from another Client.

4.9.5.2 Server Session Begin (SSB)

The Server Session Begin (SSB) signals the beginning of a Server Session. The SSB request is addressed to the WKA of the server for which a registration is beginning (e.g., FFFFFCh, name server, or FFFFFAh, management server). Receipt of the SSB request signals the intent of the requestor to subsequently identify the end of the registration with a Server Session End request. The format of the SSB CT_IU Request is shown in table 16.

A server accepting an SSB request shall respond with an Accept CT_IU as described in 4.4.3. A server unable to successfully complete an SSB request shall respond with a Reject CT_IU as described in 4.4.4.

Table 16 – SSB Request CT_IU

Item	Size (Bytes)
CT_IU Preamble	see 4.3

4.9.5.3 Server Session End (SSE)

The Server Session End (SSE) signals the end of a Server Session. The SSE request is addressed to the WKA of the server for which a session is being bracketed (e.g., FFFFFCh, name server, or FFFFFAh, management server). The format of the SSE CT_IU Request is shown in table 17.

A server accepting an SSE request shall respond with an Accept CT_IU as described in 4.4.3. A server unable to successfully complete an SSE request shall respond with a Reject CT_IU as described in 4.4.4.

Table 17 – SSE Request CT_IU

Item	Size (Bytes)
CT_IU Preamble	see 4.3

4.9.5.4 SSB and SSE Processing

SSB and SSE are issued to delineate use of a server by a Client. Each server specifies its behavior with respect to successful receipt of both Server Session Begin and End or the absence of either.

The results are not specified if an SSB or SSE request is sent in an exchange that overlaps any other CT exchange between the same Client and server.

4.9.5.5 Server Registration Sessions

Servers may allow clients to register information that is acted upon or disseminated by the server. A logically complete registration may require several requests. It may be important for the requests composing a Client registration to be applied so that the server does not act upon the results of any of the requests (e.g., issue state change notifications) until the requests composing the registration are all complete. It may also be important that the server does not act upon any requests other than those composing the registration (e.g., requests from other clients) until the requests composing the registration are all complete. Either of these requirements are called atomic registration requirements. In order for a Client to delineate the requests subject to an atomic registration requirement, the Client may communicate the commands during a single Server Session with the server. A Client requests the start of a session to a server by sending an SSB Common Request to the server. A Client declares the end of a session to a server by sending an SSE Common Request to the server. The behavior of a server with respect to commands received within a Server Session is specified for each server.

Where a server description does not include sessions, or where the action of the server is defined as being unaffected by Server Sessions, the following behavior should occur:

a) When a Server Session Begin (SSB) or Server Session End (SSE) request is addressed to the WKA of the server, it should respond with a Reject CT_IU as described in 4.4.4. with a Reject CT_IU reason code of "Invalid command code".

b) No session should be established, and access to the server from other CT Clients or management clients connected via other interfaces should not be effected in any way.

4.9.6 GS Asynchronous Notification (ASYNC)

As Fabrics scale, CT requests become more complex and require considerable time to complete the response. When a Generic Service has difficulty accepting a request without exceeding GS timeouts, the Generic Service shall reject the request with a Reject CT_IU reason code of “Unable to perform command request” and a reason code explanation of “Processing Request”. The Generic Service shall store the token from the request and notify the requesting Nx_Port upon the completion of the request with GS Asynchronous Notification.

GS Asynchronous Notification is expected to relieve repeated attempts at GS requests that fail. Instead of repeatedly failing a given request, GS Asynchronous Notification adds stability to CT request processing.

Figure 3 shows an example configuration for a GS Asynchronous Notification. An FC Trace Route request from Device C may cause the Managing Switch (Switch 4) to reject the request as seen in figure 4. The Fabric shall continue processing the request and the Managing Switch shall notify the requesting Nx_Port with GS Asynchronous Notification when the processing is completed. After receiving GS Asynchronous Notification, the requesting Nx_Port shall repeat the request with the original token and receive the stored response.

Figure 3 – Topology for GS Asynchronous Notification example

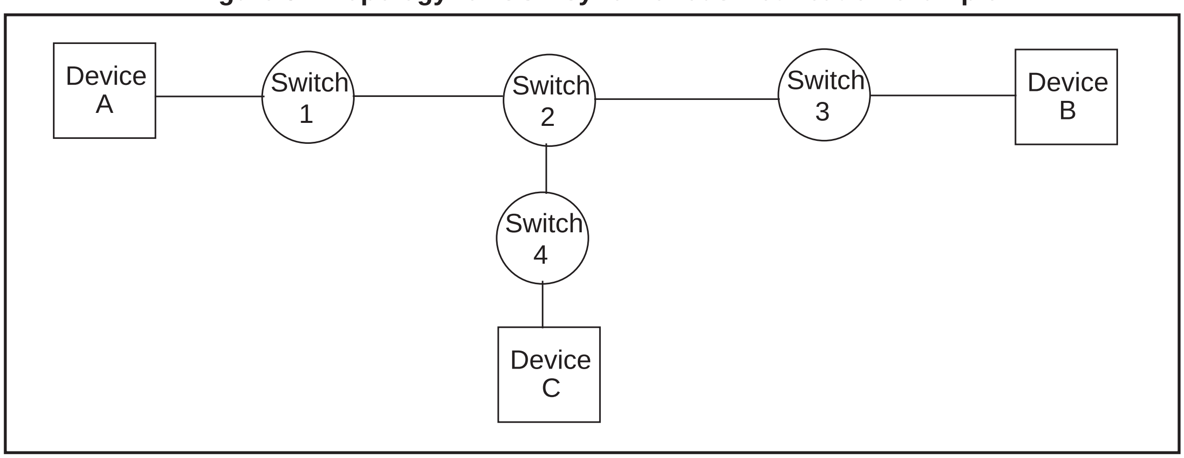
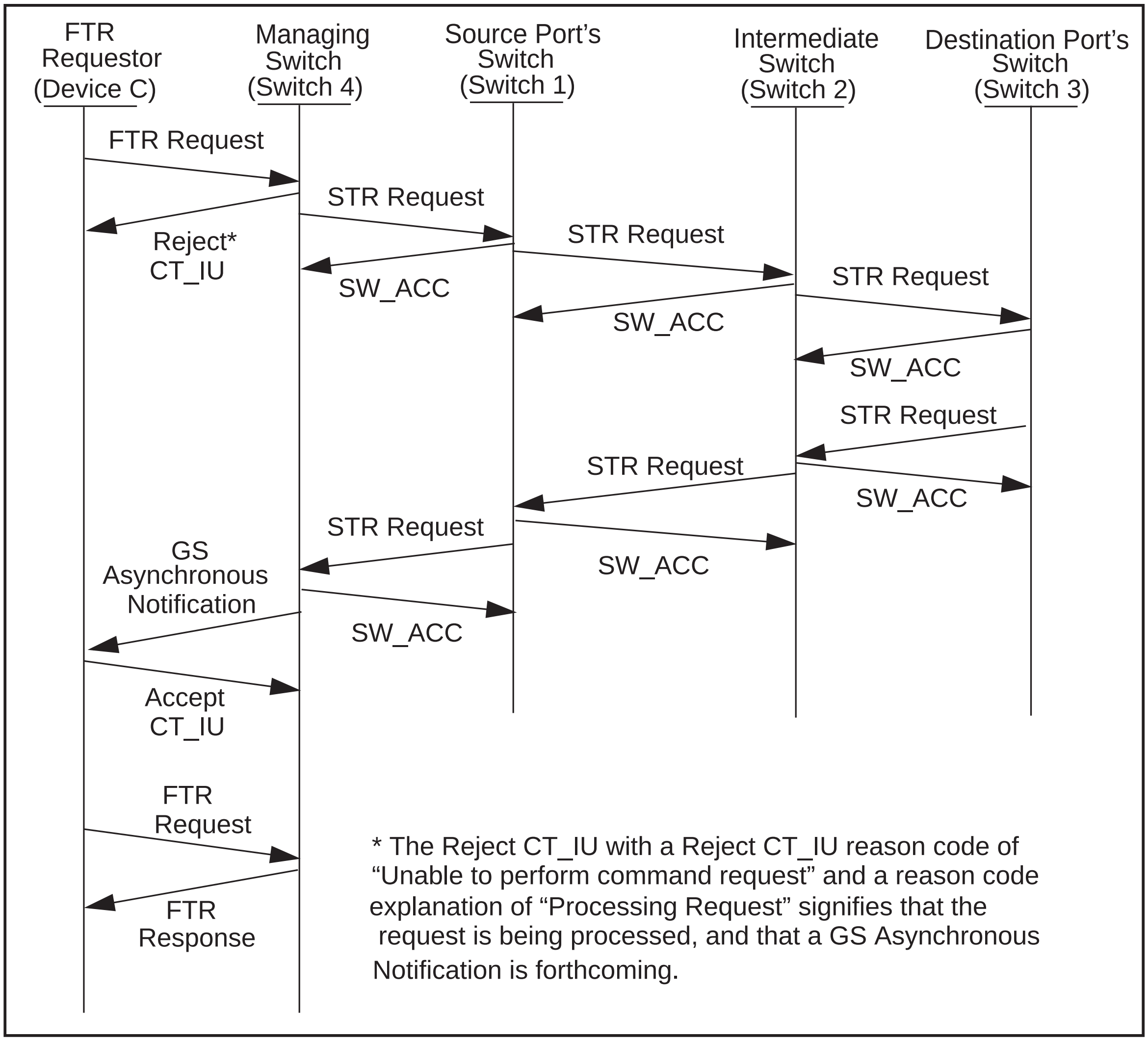


Figure 4 – FTR Processing



The Format of the GS Asynchronous Notification is shown in table 18. After receiving the GS Asynchronous Notification, an Nx_Port shall reply with an Accept CT_IU to the CT request and resend the CT request with the same token to retrieve the requested information.

Table 18 – GS Asynchronous Notification

Item	Size (Bytes)
CT_IU Preamble	see 4.3
Revision	4
Token	4

Revision: The revision shall contain a value of 01h.

Token: An identifier for a CT request provided by the requesting Nx_Port.

5 Directory Service

5.1 Overview

The Directory Service provides a means to discover information about nodes and Nx_Ports attached to a Fabric. This Service is provided through WKA FFFFFCh. The GS_Type for all Directory Services shall be set as indicated in Table 4.

This standard defines the model for requests and responses to access Directory Service information. This standard does not define the structure of this information.

Table 19 defines the GS_Subtype codes for the Directory Service.

Table 19 – Directory Service subtype values

Values	Description
01h	Obsolete[a]
02h	Name Server
03h	Obsolete[b]
04h	VE Identification Server
80h to EFh	FC-4 specific Servers
other values	Reserved

[a] Value 01h indicated the X.500 Server defined in the first publication of this standard.
[b] The IP Address Server (03h) defined in previous versions of this standard has been obsoleted.

In addition to the standard Servers, an individual FC-4 may provide its own specific Server. FC-4 based Server payloads and protocols are defined by the specific FC-4.

The consumer of a Directory Service is normally a “device driver” or some other internal layer of an operating system. Directory Service information is not normally forwarded to an application level. The information provided by a Directory Service is operational, and therefore may be constrained by the operational environment (i.e., Zone) of the node.

5.2 Name Server

5.2.1 Overview

The Name Server provides a way for Nx_Ports to register and discover Fibre Channel attributes. Once registered, the attributes are made available to requestors. Third party registration support is obsolete.

NOTE 10 – Previous versions of this standard allowed registrations to be performed by a third party, although the Name Server may refuse a third party registration for unspecified reasons.

Requests for the Name Server are carried over the Common Transport (see clause 4). Four types of requests are defined for the Name Server, as shown in table 20.

Table 20 – Name Server - request types

Command Code	Description
01xxh	Get object(s) (Query)
02xxh	Register object
03xxh	Deregister object(s)
04xxh	Delimiter based action object(s)

Table 21 lists the different Fibre Channel objects defined for the Name Server.

Table 21 – Name Server - Objects

Object Mnemonic	Object Name	Description
A	Aggregated objects	A combination of objects or object types
ID	Port Identifier	3-byte address identifier
PN	Port Name	8-byte Name_Identifier (see FC-FS-4)
NN	Node Name	8-byte Name_Identifier
CS	Class of Service	4-byte bit field, one bit per Class supported
IP	IP Address (Node) - obsolete	
IPA	Initial Process Associator - obsolete	
FT	FC-4 types	32-byte bit field, one bit per type supported
SPN	Symbolic Port Name	variable length (0 to 255-byte) field
SNN	Symbolic Node Name	variable length (0 to 255-byte) field
PT	Port Type	1-byte encoded Port Type
IPP	IP Address (Port) - obsolete	
FPN	Fabric Port Name	8-byte Name_Identifier
HA	Hard Address	3-byte address identifier
FD	FC-4 Descriptor - obsolete	
FF	FC-4 Features	128-byte array, four bits per type
PPN	Permanent Port Name	8-byte Name_Identifier

The Name Server is intended to be distributed among Switches, making the Name Server immediately available to Nx_Ports once they have successfully completed Fabric Login. However, the Name Server is not restricted or required to be part of a Fabric, and may be located in any Nx_Port. The Name Server may be made available on any Fibre Channel topology.

5.2.2 Name Server protocol

5.2.2.1 Overview,

Name Server registration, deregistration, queries, and delimiter based actions are managed through protocols containing a set of request CT_IUs and response CT_IUs supported by the Name Server.

For a Name Server request, the Name Server payload shall be transported from the requestor to the Name Server using a request CT_IU. The corresponding Name Server response is transported from the Name Server to the requestor, in the Exchange established by the requestor, using a response CT_IU.

If Zones exist within the Fabric, the Name Server shall restrict access to information in the Name Server database based on the Zone configuration (see 6.4).

5.2.2.2 CT_IU preamble values

The following values shall be set in the CT_IU preamble for Name Server requests and their responses; fields not specified here shall be set as defined in 4.3.2:

a) GS_Subtype: as indicated in table 19; and

b) Command Code: see table 22 for request command codes.

Table 22 – Name Server - Request Command Codes

Code	Mnemonic	Description	Reference subclause
0100h	GA_NXT	Get all next	5.2.5.2
0101h	GID_A	Get identifiers- scope	5.2.5.3
0112h	GPN_ID	Get Port Name	5.2.5.4
0113h	GNN_ID	Get Node Name - Port Identifier	5.2.5.5
0114h	GCS_ID	Get Class of Service	5.2.5.6
0117h	GFT_ID	Get FC-4 types	5.2.5.7
0118h	GSPN_ID	Get Symbolic Port Name	5.2.5.8
011Ah	GPT_ID	Get Port Type	5.2.5.9
011Bh	GIPP_ID	obsolete	
011Ch	GFPN_ID	Get Fabric Port Name - Port Identifier	5.2.5.10
011Dh	GHA_ID	Get Hard Address - Port Identifier	5.2.5.11

Table 22 – Name Server - Request Command Codes (Continued)

Code	Mnemonic	Description	Reference subclause
011Eh	GFD_ID	obsolete	
011Fh	GFF_ID	Get FC-4 Features - Port Identifier	5.2.5.13
0121h	GID_PN	Get Port Identifier - Port Name	5.2.5.14
012Bh	GIPP_PN	obsolete	
0131h	GID_NN	Get Port Identifiers - Node Name	5.2.5.15
0132h	GPN_NN	Get Port Names - Node Name	5.2.5.16
0135h	GIP_NN	obsolete	
0136h	GIPA_NN	obsolete	
0139h	GSNN_NN	Get Symbolic Node Name	5.2.5.17
0153h	GNN_IP	obsolete	
0156h	GIPA_IP	obsolete	
0171h	GID_FT	Get Port Identifiers - FC-4 type	5.2.5.18
0172h	GPN_FT	Get Port Names - FC-4 type	5.2.5.19
0173h	GNN_FT	Get Node Names - FC-4 type	5.2.5.20
0180h	GNN_FF	Get Node Names - FC-4 Features	5.2.5.21
0181h	GPN_FF	Get Port Names - FC-4 Features	5.2.5.22
0182h	GPN_SDFCP	Get Port Names - Simplified Discovery for FCP	5.2.5.23
01A1h	GID_PT	Get Port Identifiers - Port Type	5.2.5.24
01B1h	GID_IPP	obsolete	
01B2h	GPN_IPP	obsolete	
01C1h	GID_FPN	Get Port Identifiers - F_Port_Name	5.2.5.25
01D1h	GPPN_ID	Get Permanent Port Name - Port Identifier	5.2.5.26
01F1h	GID_FF	Get Port Identifiers - FC-4 Features	5.2.5.27
01F2h	GID_DP	Get Port Identifier - Domain/Port	5.2.5.28
0212h	RPN_ID	obsolete	
0213h	RNN_ID	Register Node Name	5.2.5.29
0214h	RCS_ID	Register Class of Service	5.2.5.30

Table 22 – Name Server - Request Command Codes (Continued)

Code	Mnemonic	Description	Reference subclause
0217h	RFT_ID	Register FC-4 types	5.2.5.31
0218h	RSPN_ID	Register Symbolic Port Name	5.2.5.32
021Ah	RPT_ID	obsolete	
021Bh	RIPP_ID	obsolete	
021Dh	RHA_ID	Register Hard Address - Port Identifier	5.2.5.33
021Eh	RFD_ID	obsolete	
021Fh	RFF_ID	Register FC-4 Features - Port Identifier	5.2.5.34
0235h	RIP_NN	obsolete	
0236h	RIPA_NN	obsolete	
0239h	RSNN_NN	Register Symbolic Node Name	5.2.5.35
0300h	DA_ID	De-register all	5.2.5.36
0400h	DBA	Delimiter Based Action[b]	5.2.2.6.2
7FF9h	SSB	Server Session Begin[a]	4.9.5.2
7FFAh	SSE	Server Session End[a]	4.9.5.3

[a] The SSB and SSE requests are two of a set of common CT_IU commands. The details of these requests is presented in 4.9. Interaction of these requests with the name server is documented in 5.2.2.7.

[b] Delimiter based actions are type length value (TLV) based CT_IU commands. The details of these commands are documented in 5.2.2.6.

5.2.2.3 Registration

A registrant submits a tuple, consisting of a primary or secondary key object along with an object to be associated with the key object. The Port Identifier is the primary key object and the Node Name the secondary key object. The secondary key shall not be used as a key object until it has been registered and associated with the primary key.

The registration requests defined for the Name Server are summarized in table 22.

Registration requests shall be processed:

a) implicitly from the Link Service Facility (i.e., Fabric F_Port);

b) from the Fabric Controller; and

c) from an Nx_Port that specifies a Port Identifier in the registration payload that matches the S_ID of the request.

The Name Server shall reject registrations from any other source.

The Name Server may reject registrations:

a) due to Name Server resource limitations; or

b) of Name Server objects associated with unassigned or unused Port Identifiers.

However, the Name Server shall support registration of all Name Server object types, once registration of a single object has been accepted for a given Port Identifier.

The Name Server shall reject all registrations of Name Server objects associated with:

a) the address identifier 00 00 00h;

b) WKA identifiers; or

c) Domain Controller identifiers (i.e., FF FC xx, where xx is any value 00h to FFh).

The Name Server may reject all registrations of Name Server objects associated with Fibre Channel addresses not used or not usable as Port Identifiers in the Fabric.

The Name Server may reject any registration requests for reasons not specified in this standard.

The Fabric may register the following objects once Fabric Login, implicit or explicit, has been successfully completed:

a) Port Type;

b) Port Identifier;

c) Port Name;

d) Node Name;

e) Class of Service; or

f) Permanent Port Name.

The Fabric may also cause the registered value of other objects to change following a successful Fabric Login. If an Nx_Port becomes logged out with the Fabric, the Fabric may de-register all objects associated with that Nx_Port.

5.2.2.4 Queries

The Name Server may reject any query requests for reasons not specified in this document.

The queries defined for the Name Server are summarized in table 22.

5.2.2.5 Deregistration

Deregistration of all Name Server objects for a given Port Identifier may be performed using the Remove All (DA_ID) request (see 5.2.5.36).

Deregistration of specific Name Server objects for a given Port Identifier may be performed by registering the NULL form of that object.

The Name Server may reject any deregistration requests for reasons not specified in this document.

5.2.2.6 Delimiter Based Actions

5.2.2.6.1 Overview

Delimiter based actions are CT_IU based Name Server commands that may be used to perform registrations, queries and deregistrations of key data /non-key data tuples.

5.2.2.6.2 Delimiter based Action Request

Each delimiter based action request shall contain:

1) a delimiter based action header indicating that the delimiter based action is a request; and

2) one or more delimiter sets (see 5.2.3.16).

The format of the request will depend upon the action requested (e.g., registration, query, deregistration) within that delimiter set.

For action types of registration and deregistration, the delimiter set shall contain:

1) one delimiter TLV that specifies the action to be performed (i.e., registration or deregistration);

2) one or more key data TLVs; and

3) one or more non-key data TLVs.

Delimiter sets shall be processed by the Name Server in the order received.

While processing a delimiter set, depending on the action specified, the Name Server shall register or deregister any information contained in the non-key TLVs with any entity that matches the condition and information specified in the delimiter TLV and key data TLVs respectively.

For an action type of query, the delimiter set shall contain:

1) one delimiter TLV that specifies the action to be performed (i.e., query);

2) one or more key data TLVs;

3) one query parameters TLV;

4) one or more non-key data TLVs.

After processing a delimiter set, the next delimiter set shall be processed if present. Once all delimiter sets have been processed, the Name Server shall respond by transmitting a delimiter based action response.

5.2.2.6.3 Delimiter based action response

In response to a delimiter based action request, the Name Server shall return a delimiter based action response. The format of the response depends on the action requested (e.g., registration, query, deregistration) within that delimiter set.

The delimiter based action response shall include:

1) A delimiter based action header indicating that the delimiter based action is a response; and

2) one delimiter set for each delimiter set in the delimiter based action request, with the order of the delimiter sets in the response the same as the order of the delimiter sets in the request.

For responses to delimiter set action types of registration and deregistration, the delimiter set shall contain:

1) One delimiter TLV that specifies the action that was requested (i.e., registration or deregistration);

2) one key data TLV for each key data TLV that was included in the request, with the order of the key data TLVs in the response the same as the order of the key data TLVs in the request. If the information contained in the request key data TLV is:

   A) recognized by the recipient, then the information contained in that key data TLV in a response shall be the same as the information in the key data TLV in the request; or

   B) unrecognized or found to be invalid by the recipient, then that key data TLV shall be a data TLV status response;

   and

3) one data TLV status response for each non-key data TLV that was included in the request with the order of the data TLVs in the response the same as the order of the data TLVs in the request. If a non-key data TLV contained information that was successfully registered with the Name Server for the specified key values, the response shall indicate that the request was successful, otherwise an appropriate error code should be provided.

For responses to delimiter set action types of query, the delimiter set shall contain:

1) one delimiter TLV that specifies the action that was requested (i.e., query);

2) one key data TLV for each key data TLV that was included in the request, with the order of the key data TLVs in the response the same as the order of the key data TLVs in the request. If the the information contained in the key data TLV of the request is:

   A) recognized by the recipient, then the information contained in that key data TLV in a response shall be the same as the information in the key data TLV in the request; or

   B) unrecognized or found to be invalid by the recipient, then that key data TLV shall be a data TLV status response;

3) one query parameters TLV; and

4) one response set of non-key data TLVs for every object in the Name Server that matches the condition and information specified in the delimiter TLV and key data TLVs respectively. Each response set shall contain one non-key data TLV for every non-key data TLV that was included in the request.

NOTE 11 – If the originator of a query request has been granted access to M objects that are currently registered with the Name Server and the request specified N non-key TLVs, then if all M objects match the condition and information specified in the delimiter TLV and key data TLVs respectively, the query response shall contain M response sets that each contain N non-key data TLVs.

5.2.2.7 Name Server Interaction with Common Requests SSB and SSE

When SSB and SSE are used in conjunction with the Name Server, an SSB request is accepted and followed by one or more specific name server registration requests. An SSE request, even in the absence of a prior SSB request, signals that the Name Server, in conjunction with the Fabric Controller, has initiative to begin the state change notification process (see 4.9.5).

In the event a Server Session End is not received after a successfully accepted Server Session Begin, the server shall initiate registered state change notification following expiration of the R_A_TOV time period or receipt of a LOGO request.

The persistence of any actions taken by the Name Server on its data in response to Client commands is independent of Server Sessions.

5.2.3 Name Server objects - formats

5.2.3.1 Use of Null values for Name Server objects

The format of the Name Server objects summarized in table 21 are described in 5.2.3. A null value is defined for each Name Server object. A null value is used in the GA_NXT Accept CT_IU to indicate an object is not registered.

If a Name Server request is received for a specific object and that object has not been registered, an Accept CT_IU with a null value for that object shall not be returned. In this case, a Reject CT_IU shall be returned with the reason code set to “Unable to perform command request” and the reason code explanation set to indicate the specified object is not registered (see 5.2.4).

5.2.3.2 Port Identifier - format

The Port Identifier is a Fibre Channel address identifier, assigned to an Nx_Port during implicit or explicit Fabric Login. The format of the Port Identifier object shall be identical to the address identifier format.

The Port Identifier serves as the unique data base key for the Name Server.

The null value for the Port Identifier is 00 00 00h.

5.2.3.3 Port Name - format

The format of the Port Name object shall be identical to the Name_Identifier format.

The null value for the Port Name object is 00 00 00 00 00 00 00 00h.

5.2.3.4 Node Name - format

The format of the Node Name object shall be identical to the Name_Identifier format.

The null value for the Node Name object is 00 00 00 00 00 00 00 00h.

5.2.3.5 Class of Service - format

The format of the Class of Service object shall be bit mapped as shown below:

Bit 0 - Class F
0 = Class F is not supported by the Port Identifier.
1 = Class F is supported by the Port Identifier.

Bit 1 - obsolete

Bit 2 - Class 2
0 = Class 2 is not supported by the Port Identifier.
1 = Class 2 is supported by the Port Identifier.

Bit 3 - Class 3
0 = Class 3 is not supported by the Port Identifier.
1 = Class 3 is supported by the Port Identifier.

Bit 4 - obsolete

Bit 5: reserved

Bit 6 - obsolete

Bits 7 to 31: reserved

The null value for the Class of Service Name Server object is 00 00 00 00h.

5.2.3.6 FC-4 types - format

The FC-4 types object indicates a set of supported FC-4 protocol type values (e.g., Fibre Channel Protocol (08h), SBCCS (18h), FC-NVMe (28h)) (see FC-FS-4). For each possible type value, this standard specifies a flag bit in the FC-4 types object that indicates whether or not the FC-4 protocol with that type value is supported. For each supported type value, the value of its flag bit in the FC-4 types object shall be one. For each unsupported type value, the value of its flag bit in the FC-4 types object shall be zero.

Table 23 – FC-4 types mapping

type value bit 4 3210	type value bit 7 6 5	type value bit 7 6 5	type value bit 7 6 5	type value bit 7 6 5	type value bit 7 6 5	type value bit 7 6 5	type value bit 7 6 5	type value bit 7 6 5
	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0 0000	wd 0 [0]	wd 1 [0]	wd 2 [0]	wd 3 [0]	wd 4 [0]	wd 5 [0]	wd 6 [0]	wd 7 [0]
0 0001	wd 0 [1]	wd 1 [1]	wd 2 [1]	wd 3 [1]	wd 4 [1]	wd 5 [1]	wd 6 [1]	wd 7 [1]
0 0010	wd 0 [2]	wd 1 [2]	wd 2 [2]	wd 3 [2]	wd 4 [2]	wd 5 [2]	wd 6 [2]	wd 7 [2]
0 0011	wd 0 [3]	wd 1 [3]	wd 2 [3]	wd 3 [3]	wd 4 [3]	wd 5 [3]	wd 6 [3]	wd 7 [3]
0 0100	wd 0 [4]	wd 1 [4]	wd 2 [4]	wd 3 [4]	wd 4 [4]	wd 5 [4]	wd 6 [4]	wd 7 [4]
0 0101	wd 0 [5]	wd 1 [5]	wd 2 [5]	wd 3 [5]	wd 4 [5]	wd 5 [5]	wd 6 [5]	wd 7 [5]
0 0110	wd 0 [6]	wd 1 [6]	wd 2 [6]	wd 3 [6]	wd 4 [6]	wd 5 [6]	wd 6 [6]	wd 7 [6]
0 0111	wd 0 [7]	wd 1 [7]	wd 2 [7]	wd 3 [7]	wd 4 [7]	wd 5 [7]	wd 6 [7]	wd 7 [7]
0 1000	wd 0 [8]	wd 1 [8]	wd 2 [8]	wd 3 [8]	wd 4 [8]	wd 5 [8]	wd 6 [8]	wd 7 [8]
0 1001	wd 0 [9]	wd 1 [9]	wd 2 [9]	wd 3 [9]	wd 4 [9]	wd 5 [9]	wd 6 [9]	wd 7 [9]
0 1010	wd 0 [10]	wd 1 [10]	wd 2 [10]	wd 3 [10]	wd 4 [10]	wd 5 [10]	wd 6 [10]	wd 7 [10]
0 1011	wd 0 [11]	wd 1 [11]	wd 2 [11]	wd 3 [11]	wd 4 [11]	wd 5 [11]	wd 6 [11]	wd 7 [11]
0 1100	wd 0 [12]	wd 1 [12]	wd 2 [12]	wd 3 [12]	wd 4 [12]	wd 5 [12]	wd 6 [12]	wd 7 [12]
0 1101	wd 0 [13]	wd 1 [13]	wd 2 [13]	wd 3 [13]	wd 4 [13]	wd 5 [13]	wd 6 [13]	wd 7 [13]
0 1110	wd 0 [14]	wd 1 [14]	wd 2 [14]	wd 3 [14]	wd 4 [14]	wd 5 [14]	wd 6 [14]	wd 7 [14]
0 1111	wd 0 [15]	wd 1 [15]	wd 2 [15]	wd 3 [15]	wd 4 [15]	wd 5 [15]	wd 6 [15]	wd 7 [15]
1 0000	wd 0 [16]	wd 1 [16]	wd 2 [16]	wd 3 [16]	wd 4 [16]	wd 5 [16]	wd 6 [16]	wd 7 [16]
1 0001	wd 0 [17]	wd 1 [17]	wd 2 [17]	wd 3 [17]	wd 4 [17]	wd 5 [17]	wd 6 [17]	wd 7 [17]
1 0010	wd 0 [18]	wd 1 [18]	wd 2 [18]	wd 3 [18]	wd 4 [18]	wd 5 [18]	wd 6 [18]	wd 7 [18]
1 0011	wd 0 [19]	wd 1 [19]	wd 2 [19]	wd 3 [19]	wd 4 [19]	wd 5 [19]	wd 6 [19]	wd 7 [19]
1 0100	wd 0 [20]	wd 1 [20]	wd 2 [20]	wd 3 [20]	wd 4 [20]	wd 5 [20]	wd 6 [20]	wd 7 [20]
1 0101	wd 0 [21]	wd 1 [21]	wd 2 [21]	wd 3 [21]	wd 4 [21]	wd 5 [21]	wd 6 [21]	wd 7 [21]
1 0110	wd 0 [22]	wd 1 [22]	wd 2 [22]	wd 3 [22]	wd 4 [22]	wd 5 [22]	wd 6 [22]	wd 7 [22]
1 0111	wd 0 [23]	wd 1 [23]	wd 2 [23]	wd 3 [23]	wd 4 [23]	wd 5 [23]	wd 6 [23]	wd 7 [23]
1 1000	wd 0 [24]	wd 1 [24]	wd 2 [24]	wd 3 [24]	wd 4 [24]	wd 5 [24]	wd 6 [24]	wd 7 [24]
1 1001	wd 0 [25]	wd 1 [25]	wd 2 [25]	wd 3 [25]	wd 4 [25]	wd 5 [25]	wd 6 [25]	wd 7 [25]
1 1010	wd 0 [26]	wd 1 [26]	wd 2 [26]	wd 3 [26]	wd 4 [26]	wd 5 [26]	wd 6 [26]	wd 7 [26]
1 1011	wd 0 [27]	wd 1 [27]	wd 2 [27]	wd 3 [27]	wd 4 [27]	wd 5 [27]	wd 6 [27]	wd 7 [27]
1 1100	wd 0 [28]	wd 1 [28]	wd 2 [28]	wd 3 [28]	wd 4 [28]	wd 5 [28]	wd 6 [28]	wd 7 [28]
1 1101	wd 0 [29]	wd 1 [29]	wd 2 [29]	wd 3 [29]	wd 4 [29]	wd 5 [29]	wd 6 [29]	wd 7 [29]
1 1110	wd 0 [30]	wd 1 [30]	wd 2 [30]	wd 3 [30]	wd 4 [30]	wd 5 [30]	wd 6 [30]	wd 7 [30]
1 1111	wd 0 [31]	wd 1 [31]	wd 2 [31]	wd 3 [31]	wd 4 [31]	wd 5 [31]	wd 6 [31]	wd 7 [31]

The mapping between a type value and its flag bit in an FC-4 types object shall be determined by these rules (see table 23):

a) The three most significant bits of the type value shall be a word number within the FC-4 types object;

   A) Word 0 contains information related to type value 00h through 1Fh;

   B) Word 1 contains information related to type value 20h through 3Fh;

   C) Word 2 contains information related to type value 40h through 5Fh;

D) Word 3 contains information related to type value 60h through 7Fh;

E) Word 4 contains information related to type value 80h through 9Fh;

F) Word 5 contains information related to type value A0h through BFh;

G) Word 6 contains information related to type value C0h through DFh; and

H) Word 7 contains information related to type value E0h through FFh.

and

b) the five least significant bits of the type value shall be the bit number of the flag bit within the FC-4 types word identified by rule (a).

A Port Identifier supporting SCSI FCP (type is 08h), ISO/IEC 8802-2 LLC/SNAP (In-order) (type is 04h) and Common Transport (type is 20h) would register 00 00 01 10 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00h as it’s FC-4 types object.

The null FC-4 types object value is set to 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00h.

5.2.3.7 Symbolic Port Name - format

The Symbolic Port Name object is of variable length, with a minimum of 0 and a maximum of 255 bytes. The contents of these bytes are not defined and shall not be restricted by the Name Server.

If a Symbolic Port Name is not registered then the Symbolic Port Name defaults to a 0 byte length object.

5.2.3.8 Symbolic Node Name - format

The Symbolic Node Name object is of variable length, with a minimum length of 0 and a maximum length of 255 bytes. The contents of these bytes are not defined and shall not be restricted by the Name Server.

If a Symbolic Node Name is not registered then the Symbolic Node Name defaults to 0 byte length object.

5.2.3.9 Port Type - format

The format of the Port Type object is a one byte value encoded as shown in table 24.

Table 24 – Port Type - encoding

Encoded value	Description
00h	Unidentified
01h	N_Port
02h	NL_Port
03h	F/NL_Port
7Fh	Nx_Port
81h	F_Port
82h	FL_Port
84h	E_Port
85h	B_Port
All others	Reserved

Port Type ‘Nx_Port’ is provided as a means to request all Port Types less than 80h. Port Type Nx_Port may only be specified in a GID_PT request CT_IU, and shall never be specified in the response to a GA_NXT or GPT_ID request CT_IU, or in an RPT_ID request CT_IU.

The null Port Type object value is set to an ‘Unidentified’ type.

5.2.3.10 Fabric Port Name - format

The format of the Fabric Port Name object shall be identical to the Name_Identifier format. The Fabric Port Name for a given Port Identifier is the Port_Name for the Fx_Port to which the PN_Port through which the Nx_Port communicates is attached.

NOTE 12 – No explicit registration command is provided for this object.

The null value for the Fabric Port Name object is 00 00 00 00 00 00 00 00h.

5.2.3.11 Hard Address - format

The format of the Hard Address object shall be identical to the format of Hard Address defined in the Discover Address (ADISC) Extended Link Service (see FC-LS-3).

The null value for the Hard Address object is 00 00 00h.

5.2.3.12 FC-4 Features - format

The format of the FC-4 Features object, as defined by the FC-4, shall be an array of 4-bit values, one for each type code value, positioned in the FC-Features object as follows (see also table 25):

a) the 5 most significant bits of the type field shall be used to identify the word for the FC-4 Features object;

   A) Word 0 contains information related to type code 00h to 07h;

   B) Word 1 contains information related to type code 08h to 0Fh;

   C) Word 2 contains information related to type code 10h to 17h; and

   D) and so forth to Word 31 that contains information related to type code F8h to FFh.

b) the 3 least significant bits of the type field shall be used to identify the position within the word for the 4-bit FC-4 Features value (see table 25); and

c) The setting and meaning of the bits within the 4-bit FC-4 Features for a specific type value are not defined by this standard.

NOTE 13 – It is intended that the FC-4 corresponding to the type value define the meaning of the 4-bit field. See Annex A for definitions known at time of publication of this standard.

The null FC-4 Features object value is all 128 bytes set to 00h.

Table 25 – FC-4 Features mapping

FC-4 type Bit 76543	FC-4 type Bit 2 1 0	FC-4 type Bit 2 1 0	FC-4 type Bit 2 1 0	FC-4 type Bit 2 1 0	FC-4 type Bit 2 1 0	FC-4 type Bit 2 1 0	FC-4 type Bit 2 1 0	FC-4 type Bit 2 1 0
	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
00000	w00 [03:00]	w00 [07:04]	w00 [11:08]	w00 [15:12]	w00 [19:16]	w00 [23:20]	w00 [27:24]	w00 [31:28]
00001	w01 [03:00]	w01 [07:04]	w01 [11:08]	w01 [15:12]	w01 [19:16]	w01 [23:20]	w01 [27:24]	w01 [31:28]
00010	w02 [03:00]	w02 [07:04]	w02 [11:08]	w02 [15:12]	w02 [19:16]	w02 [23:20]	w02 [27:24]	w02 [31:28]
00011	w03 [03:00]	w03 [07:04]	w03 [11:08]	w03 [15:12]	w03 [19:16]	w03 [23:20]	w03 [27:24]	w03 [31:28]
00100	w04 [03:00]	w04 [07:04]	w04 [11:08]	w04 [15:12]	w04 [19:16]	w04 [23:20]	w04 [27:24]	w04 [31:28]
00101	w05 [03:00]	w05 [07:04]	w05 [11:08]	w05 [15:12]	w05 [19:16]	w05 [23:20]	w05 [27:24]	w05 [31:28]
00110	w06 [03:00]	w06 [07:04]	w06 [11:08]	w06 [15:12]	w06 [19:16]	w06 [23:20]	w06 [27:24]	w06 [31:28]
00111	w07 [03:00]	w07 [07:04]	w07 [11:08]	w07 [15:12]	w07 [19:16]	w07 [23:20]	w07 [27:24]	w07 [31:28]
01000	w08 [03:00]	w08 [07:04]	w08 [11:08]	w08 [15:12]	w08 [19:16]	w08 [23:20]	w08 [27:24]	w08 [31:28]
01001	w09 [03:00]	w09 [07:04]	w09 [11:08]	w09 [15:12]	w09 [19:16]	w09 [23:20]	w09 [27:24]	w09 [31:28]
01010	w10 [03:00]	w10 [07:04]	w10 [11:08]	w10 [15:12]	w10 [19:16]	w10 [23:20]	w10 [27:24]	w10 [31:28]
01011	w11 [03:00]	w11 [07:04]	w11 [11:08]	w11 [15:12]	w11 [19:16]	w11 [23:20]	w11 [27:24]	w11 [31:28]
01100	w12 [03:00]	w12 [07:04]	w12 [11:08]	w12 [15:12]	w12 [19:16]	w12 [23:20]	w12 [27:24]	w12 [31:28]
01101	w13 [03:00]	w13 [07:04]	w13 [11:08]	w13 [15:12]	w13 [19:16]	w13 [23:20]	w13 [27:24]	w13 [31:28]
01110	w14 [03:00]	w14 [07:04]	w14 [11:08]	w14 [15:12]	w14 [19:16]	w14 [23:20]	w14 [27:24]	w14 [31:28]
01111	w15 [03:00]	w15 [07:04]	w15 [11:08]	w15 [15:12]	w15 [19:16]	w15 [23:20]	w15 [27:24]	w15 [31:28]
10000	w16 [03:00]	w16 [07:04]	w16 [11:08]	w16 [15:12]	w16 [19:16]	w16 [23:20]	w16 [27:24]	w16 [31:28]
10001	w17 [03:00]	w17 [07:04]	w17 [11:08]	w17 [15:12]	w17 [19:16]	w17 [23:20]	w17 [27:24]	w17 [31:28]
10010	w18 [03:00]	w18 [07:04]	w18 [11:08]	w18 [15:12]	w18 [19:16]	w18 [23:20]	w18 [27:24]	w18 [31:28]
10011	w19 [03:00]	w19 [07:04]	w19 [11:08]	w19 [15:12]	w19 [19:16]	w19 [23:20]	w19 [27:24]	w19 [31:28]
10100	w20 [03:00]	w20 [07:04]	w20 [11:08]	w20 [15:12]	w20 [19:16]	w20 [23:20]	w20 [27:24]	w20 [31:28]
10101	w21 [03:00]	w21 [07:04]	w21 [11:08]	w21 [15:12]	w21 [19:16]	w21 [23:20]	w21 [27:24]	w21 [31:28]
10110	w22 [03:00]	w22 [07:04]	w22 [11:08]	w22 [15:12]	w22 [19:16]	w22 [23:20]	w22 [27:24]	w22 [31:28]
10111	w23 [03:00]	w23 [07:04]	w23 [11:08]	w23 [15:12]	w23 [19:16]	w23 [23:20]	w23 [27:24]	w23 [31:28]
11000	w24 [03:00]	w24 [07:04]	w24 [11:08]	w24 [15:12]	w24 [19:16]	w24 [23:20]	w24 [27:24]	w24 [31:28]
11001	w25 [03:00]	w25 [07:04]	w25 [11:08]	w25 [15:12]	w25 [19:16]	w25 [23:20]	w25 [27:24]	w25 [31:28]
11010	w26 [03:00]	w26 [07:04]	w26 [11:08]	w26 [15:12]	w26 [19:16]	w26 [23:20]	w26 [27:24]	w26 [31:28]
11011	w27 [03:00]	w27 [07:04]	w27 [11:08]	w27 [15:12]	w27 [19:16]	w27 [23:20]	w27 [27:24]	w27 [31:28]
11100	w28 [03:00]	w28 [07:04]	w28 [11:08]	w28 [15:12]	w28 [19:16]	w28 [23:20]	w28 [27:24]	w28 [31:28]
11101	w29 [03:00]	w29 [07:04]	w29 [11:08]	w29 [15:12]	w29 [19:16]	w29 [23:20]	w29 [27:24]	w29 [31:28]
11110	w30 [03:00]	w30 [07:04]	w30 [11:08]	w30 [15:12]	w30 [19:16]	w30 [23:20]	w30 [27:24]	w30 [31:28]
11111	w31 [03:00]	w31 [07:04]	w31 [11:08]	w31 [15:12]	w31 [19:16]	w31 [23:20]	w31 [27:24]	w31 [31:28]

5.2.3.13 Permanent Port Name - format

The Permanent Port Name is a name identifier associated with a PN_Port and its N_Port_IDs and shall be the F_Port Name of the F_Port that is attached to the PN_Port.

The format of the Permanent Port Name object shall be identical to the Name_Identifier format.

NOTE 14 – No explicit registration command is provided for this object.

The null value for the Permanent Port Name object is 00 00 00 00 00 00 00 00h.

5.2.3.14 Delimiter based action - format

The format of a delimiter based action is shown in table 26.

Each delimiter based action shall contain a delimiter based action header and one or more variable length delimiter sets.

Table 26 – Delimiter based action - format

Item	Size (Bytes)
CT_IU preamble	see 4.3
Delimiter based action header	see 5.2.3.15
Delimiter set 1	see 5.2.3.16
Delimiter set n	see 5.2.3.16

5.2.3.15 Delimiter based action header - format

The format of the delimiter based action header is shown in table 27.

The format of the delimiter flags field is shown in table 28

The Total length field shall be set to the total length in words of the delimiter based action. This includes the delimiter based action header and all delimiter sets.

Table 27 – Delimiter based action header - format

Item	Size (Bytes)
Delimiter flags	1
Reserved	1
Total length (words)	2

Table 28 – Delimiter flags - format

Bit Position	Description
7	Response. This bit shall be set to one if the delimiter based action contains response delimiter sets (i.e., Bit 31 of the action flags field in the delimiter TLV is set to one, see 5.2.3.17). If this bit is set to one, the delimiter based action shall be referred to as a delimiter based action response. This bit shall be set to zero when the delimiter based action does not contain response delimiter sets (i.e., Bit 31 of the delimiter TLV is set to zero). If this bit is set to zero, the delimiter based action shall be referred to as a delimiter based action request.
6 to 1	Reserved
0	Error. If the response bit (i.e., bit 7) is set to zero, this bit shall be set to zero. If the response bit is set to one, this bit shall be set to one if any of the TLVs in any of the delimiter sets contained in the delimiter based action have a status other than no error (i.e., 00 00 00 00). This bit shall be set to zero if all TLVs in any of the delimiter sets contained in the delimiter based action have a status of no error (i.e., 00 00 00 00).

5.2.3.16 Delimiter set - format

The format of the delimiter set shall depend on the type of action specified:

The format of the delimiter set used when an action type of register or deregister is specified is shown in table 29.

Each delimiter set used to perform register or deregister shall contain:

1) one delimiter TLV;
2) one or more key data TLVs; and
3) one or more non-key data TLVs.

All key data TLVs must precede non-key data TLVs.

Table 29 – Delimiter set - register and deregister format

Item	Size (Bytes)
Delimiter TLV	8
Key data TLV 1	see 5.2.3.18
Key data TLV m	see 5.2.3.18
Non-key data TLV 1	see 5.2.3.18
Non-key data TLV n	see 5.2.3.18

The format of the delimiter set used when an action Type of query is specified is shown in table 30.

Each delimiter set used to perform query shall contain:

1) one delimiter TLV;
2) one or more key data TLVs;
3) one query parameters TLV; and
4) one or more non-key data TLVs.

All key data TLVs shall precede the query parameters TLV. The query parameters TLV shall precede any non-key TLVs.

Table 30 – Delimiter set - query format

Item	Size (Bytes)
Delimiter TLV	8
Key data TLV 1	see 5.2.3.18
Key data TLV m	see 5.2.3.18
Query parameters TLV	6
Non-key data TLV 1	see 5.2.3.18
Non-key data TLV n	see 5.2.3.18

5.2.3.17 Delimiter TLV - format

The format of the delimiter TLV is shown in table 31.

The type field shall be set 60h.

The length field shall be set to the length in words of the delimiter set including the delimiter TLV and all data TLVs.

Table 31 – Delimiter TLV format

Item	Size (Bytes)
Type (object mnemonic Type code = 60h)	2
Length (words)	2
Action code	1
Key logical relationship flags	1
Action Flags	2

The format of the action flags field is shown in table 32.

Table 32 – Action flags - format

Bit Position	Description
31 to 24	Action code. This field indicates the action being requested within the delimiter set: - 0000 0001 = Query - 0000 0010 = Register - 0000 0011 = De-Register - 0000 0100 = Vendor specific
23	Key logical relationship flags - Logical AND. This bit shall be set to zero when an action type of register or De-Register is specified. This bit may be set to one when an action type of query or vendor specific is specified. When an action type of query is specified and this bit is set to one and the logical NOT bit is set to zero, the device performing the query is requesting that a logical AND be performed against all key data TLVs and only those entities registered with the Name Server that match all key data TLV(s) be returned. When an action type of query is specified and this bit is set to one and the logical NOT bit is set to one, the device performing the Query is requesting that a logical NAND be performed against all key data TLVs and only those entities registered with the Name Server that do not match all key data TLV(s) be returned. When an action type of Vendor Specific is specified, the meaning of this bit is undefined. The logical AND and logical OR bits are mutually exclusive and shall not be set to one at the same time. If both the AND and OR bits are set to one, the Name Server shall reject the delimiter set by returning a single data TLV status response TLV with a reason code / reason code explanation of logical error (03)/ No additional explanation (00).
22	Key logical relationship flags - Logical OR. This bit shall be set to zero when an action type of register or de-register is specified. This bit may be set to one when an action type of query or vendor specific is specified. When an action type of query is specified and this bit is set to one and the logical NOT bit is set to zero, the device performing the query is requesting that a logical OR be performed against all key data TLVs and those entities registered with the Name Server that match any key data TLV be returned. When an action type of query is specified and this bit is set to one and the logical NOT bit is set to one, the device performing the query is requesting that a logical NOR be performed against all key data TLVs and only those entities registered with the Name Server that do not match any key data TLV be returned. When an action type of vendor specific is specified, the meaning of this bit is undefined. The logical AND and logical OR bits are mutually exclusive and shall not be set to one at the same time. If both the AND and OR bits are set to one, the Name Server shall reject the delimiter set by returning a single data TLV status response TLV with a reason code / reason code explanation of logical error (03)/ No additional explanation (00).
21	Key logical relationship flags - Logical NOT. This bit shall be set to zero when an action type of register or de-register is specified. This bit may be set to one when an action type of query or vendor specific is specified and either the logical AND or logical OR bits are set. The meaning of this bit is defined in the logical AND and logical OR definitions.
20 to 16	Key logical relationship flags - Reserved

Table 32 – Action flags - format(Continued)

Bit Position	Description
15	Action Flags - Error All. When this bit is set to one, the device is requesting that if any of the key data or non-key data TLVs contain information that is either unrecognized or cannot be processed for any reason; the command defined in the delimiter set should fail completely and no information will be registered, deregistered or returned from the name server other than error information. When this bit is set to zero, the device is requesting that if any of the key data or non-key data TLVs contain information that is either unrecognized or cannot be processed for any reason; the command defined in the delimiter set should complete on a best effort basis. The key data or non-key data TLVs that contained unrecognizable information shall be indicated by an error code in the command response.
14 to 0	Action Flags - Reserved

5.2.3.18 Data TLV - format

The format of the data TLV is shown in table 33.

Table 33 – Data TLV - format

Item	Size (Bytes)
Type	2
Length (words)	2
Value	variable

The format of the type field is shown in table 34

Table 34 – Type Field - format

Bit Position	Description
15	Key. The key bit may be set to one if the object mnemonic type is allowed to be a Registration key (see table 33). When the key bit is set to one, the data TLV is referred to as a key data TLV. When the key bit is set to zero, the data TLV is referred to as a non-key data TLV. Key data TLVs must be listed first in a delimiter set followed by non-Key data TLVs if present. If a key data TLV is listed after a non-key data TLV, the Name Server shall reject the delimiter set by returning a single data TLV status response TLV with a reason code / reason code explanation of logical Error (03)/ No additional explanation (00).
14 to 0	object mnemonic type code (see table 35)

The length field shall contain the length in words of the entire data TLV

The Value field shall contain the information contained in the object Name column for the type code specified. This information shall be word aligned with the least significant byte(s) padded with zeros if necessary.

5.2.3.19 Object mnemonic type code - format

The type field of data TLVs used in delimiter based action commands shall be set to one of the object mnemonic type codes defined in table 35. Only those object mnemonic type codes containing a Y in the “Reg Key Allowed” field may be used when a data TLVs key bit is set to one. Object mnemonic type codes that do not contain a Y in the “Reg Key Allowed” field shall not be used when a data TLVs key bit is set to one. If the data TLV key bit is set to one and the object mnemonic type code does not contain a Y in the “Reg Key Allowed” field, the Name Server shall reject the delimiter set by returning a single data TLV status response TLV with a reason code / reason code explanation of logical Error (03)/ No additional explanation (00).

Table 35 – Object mnemonic Type Codes

Code	object mnemonic	Reg Key Allowed	object Name	Description
01h	ID	Y	Port Identifier	3-byte address identifier
02h	PN	Y	Port Name	8-byte Name_Identifier
03h	NN	Y	Node Name	8-byte Name_Identifier
04h	CS	N	Class of Service	4-byte bit field, one bit per Class supported
05h	FT	N	FC-4 types	32-byte bit field, one bit per type supported
06h	SPN	N	Symbolic Port Name	variable length (0 to 255-byte) field
07h	SNN	N	Symbolic Node Name	variable length (0 to 255-byte) field
08h	PT	N	Port Type	1-byte encoded Port Type
09h	FPN	N	Fabric Port Name	8-byte Name_Identifier
0Ah	HA	N	Hard Address	3-byte address identifier
0Bh	FF	N	FC-4 Features	128-byte array, four bits per type
0Ch	PPN	N	Permanent Port Name	8-byte Name_Identifier
60h	Delimiter	N/A	Delimiter TLV	Type length Value formatted data (see 5.2.3.17)
61h	Query Parameters	N	Query Parameters TLV	Type length Value formatted data (see 5.2.3.20)
62h	Data TLV Status	N	data TLV status response	Type length Value formatted data (see 5.2.3.21)
70h to 7Fh	Vendor Specific	Y	Vendor Specific	Vendor Specific

5.2.3.20 Query parameters TLV - format

The format of the query parameters TLV is shown in table 36.

Table 36 – Query parameters TLV - format

Item	Size (Bytes)
Type	2
Query parameters length (words)	2
Reserved	1
Number of Name Server entries requested	3
Reserved	1
Starting N_Port_ID	3
Reserved	1
Number of Name Server entries available	3
Reserved	1
Number of Name Server entries returned	3
Reserved	1
Last N_Port_ID returned	3

The format of the type field is shown in table 37.

Table 37 – Type field - format

Bit Position	Description
15	Reserved
14 to 0	Object mnemonic type Code = 61h (see table 35)

The query parameters length field shall be set to the length in words of the entire query parameters TLV including the type and the query parameters length field.

The contents of the remaining fields shall depend upon the source of the delimiter based action.

If the query parameters TLV is being used in a delimiter based action request:

a) The Number of entries requested field shall be set to the maximum number of Name Server entries that the device is requesting to be returned;

b) The starting N_Port_ID field shall be set to the first N_Port_ID value that should be returned by the Name Server; and

c) The number of Name Server entries available, number of Name Server entries returned and last N_Port_ID returned fields shall be reserved.

If the query parameters TLV is being used in a delimiter based action response:

a) The number of entries requested field shall be set to the value specified in the delimiter based action request;

b) The starting N_Port_ID field shall be set to the value specified in the delimiter based action request;

c) The number of Name Server entries available shall be set to the total number of N_Port_IDs available to the requesting device.

d) The number of Name Server entries returned shall be set to the number of response sets returned by the Name Server. A response set shall contain one non-key data TLV for every non-key data TLV in the delimiter set that contained the query action type.

e) The last N_Port_ID returned field shall be set to the N_Port_ID associated with the last response set returned by the Name Server.

5.2.3.21 Data TLV status response - format

The format of the data TLV status response is shown in table 38.

Table 38 – Data TLV status response - format

Item	Size (Bytes)
Type	2
Length (words)	2
Reserved	2
Reason Code	1
Reason Code Explanation	1

The format of the type field is shown in table 39.

Table 39 – Data TLV status response type field - format

Bit Position	Description
15	Reserved
14 to 0	Object mnemonic type Code = 62h (see table 35)

The length field shall be set to 02h.

The Reason Code field shall be set to an appropriate reason code (see table 10).

The Reason Code Explanation field shall be set to an appropriate reason code explanation (see table 40).

5.2.4 Reason code explanations

A Reject CT_IU (see 4.4.4) shall notify the requestor that the request has failed. The first error condition encountered shall be the error reported by the Reject CT_IU.

If a valid Name Server request is not received, the request is rejected with a reason code of “Invalid command code” and a reason code explanation of “No additional explanation”.

If a Name Server request is rejected with a reason code of ‘Unable to perform command request’, then one of the reason code explanations shown in table 40 is returned.

Table 40 – Reject CT_IU Reason code explanations

Encoded value	Description
00h	No additional explanation
01h	Port Identifier not registered
02h	Port Name not registered
03h	Node Name not registered
04h	Class of Service not registered
05h	Obsolete
06h	Obsolete
07h	FC-4 types not registered
08h	Symbolic Port Name not registered
09h	Symbolic Node Name not registered
0Ah	Port Type not registered
0Bh	Obsolete

Table 40 – Reject CT_IU Reason code explanations(Continued)

Encoded value	Description
0Ch	Fabric Port Name not registered
0Dh	Hard Address not registered
0Eh	Obsolete
0Fh	FC-4 Features not registered
10h	Access denied
11h	Unacceptable Port Identifier
12h	Data base empty
13h	No object registered within the specified scope
14h	Domain ID not present
15h	Port number not present
16h	No device attached
Others	Reserved

The use of these codes is further defined as follows:

a) If a Name Server request is rejected by the Name Server because of the identity of the requestor, then the Reject CT_IU reason code shall be ‘Unable to perform command request’, with a reason code explanation of ‘Access denied’;

b) If a Name Server Query request is rejected by the Name Server because no Name Server entries exist, then the Reject CT_IU reason code shall be ‘Unable to perform command request’, with a reason code explanation of ‘data base empty’;

c) If a Name Server Query request other than GA_NXT is rejected by the Name Server because the object specified in the request is not found in the Name Server data base (within the specified scope, in the case of GID_PT), then the Reject CT_IU reason code shall be ‘Unable to perform command request’, with a reason code explanation that indicates the specified object is not registered;

d) If a Name Server GID_A Query request is rejected by the Name Server because the requested information is not found within the specified Domain_ID Scope, then the Reject CT_IU reason code shall be ‘Unable to perform command request’, with a reason code explanation of ‘No object registered within the specified scope’;

e) If a Name Server Registration request is rejected by the Name Server because the Port Identifier fails registration, then the Reject CT_IU reason code shall be ‘Unable to perform command request’, with a reason code explanation of ‘Unacceptable Port Identifier’; or

f) Additional uses may be defined for specific Name Server requests.

5.2.5 Commands

5.2.5.1 Overview

The commands defined for the Name Server are summarized in table 22.

5.2.5.2 Query - Get all next (GA_NXT)

The GA_NXT is used by a requestor to obtain Name Server objects associated with a specific Port Identifier. The Name Server shall return Name Server objects, not for the supplied Fibre Channel address identifier, but for the next higher valued Port Identifier, registered with the Name Server. If there are no registered Port Identifier higher valued than the value in the GA_NXT request CT_IU, then the Name Server shall return the Name Server objects for the lowest registered Port Identifier. If there are no registered Name Server objects, then the Name Server shall reject the GA_NXT request. Fibre Channel address identifiers are treated as 24 bit unsigned entities for the purposes of comparison.

NOTE 15 – No information is returned for WKAs or Domain Controller addresses.

To obtain all information on a specific Port Identifier a requestor may set the value of the Port Identifier in the request CT_IU to be one less than the Port Identifier for which information is sought.

The GA_NXT request may be used to find all registered Port Identifiers in the Fabric, by reissuing the GA_NXT request, using the Port Identifier obtained from the Accept CT_IU, and then stopping when the initially used Port Identifier threshold is recrossed.

The format of the GA_NXT request CT_IU is shown in table 41. The requestor supplies a Port Identifier using the format in 5.2.3.2, and the Name Server returns Name Server objects for the next higher valued Port Identifier.

Table 41 – GA_NXT Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Reserved	1
Port Identifier	3

The format of the Accept CT_IU to a GA_NXT request is shown in table 42. The format of the various objects returned is defined in 5.2.3.

The Port Type field returns the registered value for the Port Type, or the null value if no Port Type is registered for the Port Identifier.

The Port Identifier field indicates the Name Server entry for which association and other objects are returned.

The Port Name field returns the registered value for the Port Name, or the null value if no Port Name is registered for the Port Identifier.

The length of Symbolic Port Name field shall contain a single byte unsigned value indicating the size of the variable length Symbolic Port Name.

The Symbolic Port Name field returns the registered value for the Symbolic Port Name, or the null value if no Symbolic Port Name is registered for the Port Identifier.

The Node Name field returns the registered value for the Node Name, or the null value if no Node Name is registered for the Port Identifier.

The length of Symbolic Node Name field shall contain a single byte unsigned value indicating the size of the variable length Symbolic Node Name.

The Symbolic Node Name field returns the registered value for the Symbolic Node Name, or the null value if no Symbolic Node Name is registered for the Port Identifier.

The Class of Service field returns the registered value for the Class of Service object, or the null value if no Class of Service object is registered for the Port Identifier.

The FC-4 types object field returns the registered value for the FC-4 types object, or the null value if no FC-4 types object is registered for the Port Identifier.

The Fabric Port Name field returns the registered value for the Fabric Port Name, or the null value if no Fabric Port Name is registered for the Port Identifier.

The Hard Address field returns the registered value for the Hard Address, or the null value if no Hard Address is registered for the Port Identifier.

Table 42 – Accept CT_IU to GA_NXT Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Port Type	1
Port Identifier	3
Port Name	8
Length of Symbolic Port Name (m)	1
Symbolic Port Name	m
Reserved	255-m
Node Name	8
Length of Symbolic Node Name (n)	1
Symbolic Node Name	n
Reserved	255-n
Obsolete	8
Obsolete	16
Class of Service	4
FC-4 types	32
Obsolete	16
Fabric Port Name	8
Reserved	1
Hard Address	3

5.2.5.3 Query - Get identifiers (GID_A)

The Name Server shall, if it receives a GID_A request, return identifiers for the specified scope. The format of the GID_A request CT_IU is shown in table 43. The requestor supplies a Domain_ID Scope that defines the scope for which identifiers are sought.

NOTE 16 – The identifiers returned by this request are not Port Identifier objects. The intended purpose of this command is to allow the Name Server user to determine which Domains and Areas are available for use in the Scope field of other Queries.

The Domain_ID Scope field specifies the scope of the request. If the Domain_ID Scope field is zero, the Name Server shall return a list of Domain_IDs corresponding to registered Port Identifiers. If the Domain_ID Scope field is non-zero, the Name Server shall return a list of Domain_ID and Area_ID pairs within the Domain specified by the Domain_ID Scope corresponding to registered Port Identifiers.

Table 43 – GID_A Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Reserved	1
Domain_ID Scope	1
Reserved	2

The formats of the reply Accept CT_IU to a GID_A request are shown in table 44 and table 45.

One or more identifiers are returned. Each returned identifier is preceded by an 8 bit Control field. The format of the Control field is:

a) Bit 7 is set to zero if the identifier following the Control field is not the last identifier to be returned by the Accept CT_IU; the bit is set to one if the identifier following the Control field is the last identifier returned by the Accept CT_IU; and

b) Bits 6 to 0 are reserved.

The identifiers may be returned in any order and the order may be different for every request even if the same identifiers are returned and the requestor is the same.

Table 44 – Accept CT_IU to GID_A Request, Domain_ID Scope is zero

Item	Size (Bytes)
CT_IU preamble	see 4.3
Control (0 r r r r r r r)	1
Domain_ID #1	1
Reserved	2
...	
Control (1 r r r r r r r)	1
Domain_ID #n	1
Reserved	2

Table 45 – Accept CT_IU to GID_A Request, Domain_ID Scope is non-zero

Item	Size (Bytes)
CT_IU preamble	see 4.3
Control (0 r r r r r r r)	1
Requested Domain_ID	1
Area_ID #1	1
Reserved	1
...	
Control (1 r r r r r r r)	1
Requested Domain_ID	1
Area_ID #n	1
Reserved	1

5.2.5.4 Query - Get Port Name (GPN_ID)

The Name Server shall, if it receives a GPN_ID request, return the registered Port Name object for the specified Port Identifier. The format of the GPN_ID request CT_IU is shown in table 46. The requestor supplies the Port Identifier for which the Port Name is sought.

Table 46 – GPN_ID Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Reserved	1
Port Identifier	3

The format of the Accept CT_IU to a GPN_ID request is shown in table 47.

The Port Name field returns the registered value for the Port Name.

Table 47 – Accept CT_IU to GPN_ID Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Port Name	8

5.2.5.5 Query - Get Node Name (GNN_ID)

The Name Server shall, if it receives a GNN_ID request, return the registered Node Name object for the specified Port Identifier. The format of the GNN_ID request CT_IU is shown in table 48. The requestor supplies the Port Identifier for which the Node Name is sought.

Table 48 – GNN_ID Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Reserved	1
Port Identifier	3

The format of the Accept CT_IU to a GNN_ID request is shown in table 49.

The Node Name field returns the registered value for the Port Identifier.

Table 49 – Accept CT_IU to GNN_ID Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Node Name	8

5.2.5.6 Query - Get Class of Service (GCS_ID)

The Name Server shall, if it receives a GCS_ID request, return the registered Class of Service object for the specified Port Identifier. The format of the GCS_ID request CT_IU is shown in table 50. The requestor supplies the Port Identifier for which the Class of Service object is sought.

Table 50 – GCS_ID Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Reserved	1
Port Identifier	3

The format of the Accept CT_IU to a GCS_ID request is shown in table 51.

The Class of Service field returns the registered value for the Class of Service object.

Table 51 – Accept CT_IU to GCS_ID Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Class of Service	4

5.2.5.7 Query - Get FC-4 types (GFT_ID)

The Name Server shall, if it receives a GFT_ID request, return the registered FC-4 types object for the specified Port Identifier. The format of the GFT_ID request CT_IU is shown in table 52. The requestor supplies the Port Identifier for which the FC-4 types object is sought.

Table 52 – GFT_ID Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Reserved	1
Port Identifier	3

The format of the Accept CT_IU to a GFT_ID request is shown in table 53.

The FC-4 types field (see 5.2.3.6) returns the registered value for the FC-4 types.

Table 53 – Accept CT_IU to GFT_ID Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
FC-4 types	32

5.2.5.8 Query - Get Symbolic Port Name (GSPN_ID)

The Name Server shall, if it receives a GSPN_ID request, return the registered Symbolic Port Name for the specified Port Identifier. The format of the GSPN_ID request CT_IU is shown in table 54. The requestor supplies the Port Identifier for which the Symbolic Port Name is sought.

Table 54 – GSPN_ID Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Reserved	1
Port Identifier	3

The format of the Accept CT_IU to a GSPN_ID request is shown in table 55.

The Name length field shall contain a single byte unsigned value indicating the size of the variable length Symbolic Port Name.

The Symbolic Port Name field returns the registered Symbolic Port Name for the specified Port Identifier.

Table 55 – Accept CT_IU to GSPN_ID Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Name length (m)	1
Symbolic Port Name	m
Reserved	255-m

5.2.5.9 Query - Get Port Type (GPT_ID)

The Name Server shall, if it receives a GPT_ID request, return the registered Port Type for the specified Port Identifier. The format of the GPT_ID request CT_IU is shown in table 56. The requestor supplies the Port Identifier for which the Port Type is sought.

Table 56 – GPT_ID Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Reserved	1
Port Identifier	3

The format of the Accept CT_IU to a GPT_ID request is shown in table 57.

The Port Type field returns the registered Port Type for the specified Port Identifier.

Table 57 – Accept CT_IU to GPT_ID Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Port Type	1
Reserved	3

5.2.5.10 Query - Get Fabric Port Name (GFPN_ID)

The Name Server shall, if it receives a GFPN_ID request, return the registered Fabric Port Name object for the specified Port Identifier. The format of the GFPN_ID request CT_IU is shown in table 58. The requestor supplies the Port Identifier for which the Fabric Port Name is sought.

Table 58 – GFPN_ID Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Reserved	1
Port Identifier	3

The format of the Accept CT_IU to a GFPN_ID request is shown in table 59.

The Fabric Port Name field returns the registered value for the Fabric Port Name.

Table 59 – Accept CT_IU to GFPN_ID Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Fabric Port Name	8

5.2.5.11 Query - Get Hard Address (GHA_ID)

The Name Server shall, if it receives a GHA_ID request, return the registered Hard Address object for the specified Port Identifier. The format of the GHA_ID request CT_IU is shown in table 60. The requestor supplies the Port Identifier for which the Hard Address is sought.

Table 60 – GHA_ID Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Reserved	1
Port Identifier	3

The format of the Accept CT_IU to a GHA_ID request is shown in table 61.

The Hard Address field returns the registered value for the Hard Address.

Table 61 – Accept CT_IU to GHA_ID Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Reserved	1
Hard Address	3

~~5.2.5.12 Query - Get FC-4 Descriptors (GFD_ID) - obsolete~~

~~The GFD_ID CT_IU request is obsolete.~~

5.2.5.13 Query - Get FC-4 Features (GFF_ID)

The Name Server shall, if it receives a GFF_ID request, return the registered FC-4 Features object for the specified Port Identifier. The format of the GFF_ID request CT_IU is shown in table 62. The requestor supplies the Port Identifier for which the FC-4 Features object is sought.

Table 62 – GFF_ID Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Reserved	1
Port Identifier	3

The format of the Accept CT_IU to a GFF_ID request is shown in table 63.

The FC-4 Features field (see 5.2.3.12) returns the registered value for the FC-4 Features.

Table 63 – Accept CT_IU to GFF_ID Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
FC-4 Features	128

5.2.5.14 Query - Get Port Identifier (GID_PN)

The Name Server shall, if it receives a GID_PN request, return the Port Identifier associated with the specified Port Name. The format of the GID_PN request CT_IU is shown in table 64. The requestor supplies the Port Name for which the Port Identifier is sought.

Table 64 – GID_PN Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Port Name	8

The format of the Accept CT_IU to a GID_PN request is shown in table 65.

The Port Identifier field returns the registered Port Identifier value for the specified Port Name.

Table 65 – Accept CT_IU to GID_PN Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Reserved	1
Port Identifier	3

5.2.5.15 Query - Get Port Identifiers (GID_NN)

The Name Server shall, if it receives a GID_NN request, return all Port Identifiers registered for the specified Node Name. The format of the GID_NN request CT_IU is shown in table 66. The requestor supplies the Node Name for which associated Port Identifiers are sought.

Table 66 – GID_NN Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Node Name	8

The format of the Accept CT_IU to a GID_NN request is shown in table 67.

One or more Port Identifiers are returned. Each returned Port Identifier is preceded by an 8 bit Control field. The format of the Control field is:

a) Bit 7 is set to zero if the Port Identifier following the Control field is not the last Port Identifier to be returned by the Accept CT_IU; the bit is set to one if the Port Identifier following the Control field is the last Port Identifier returned by the Accept CT_IU; and

b) Bits 6 to 0 are reserved.

Table 67 – Accept CT_IU to GID_NN Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Control (0 r r r r r r r)	1
Port Identifier #1	3
...	
Control (1 r r r r r r r)	1
Port Identifier #n	3

The Port Identifiers may be returned in any order and the order may be different for every request even if the same Port Identifiers are returned and the requestor is the same.

5.2.5.16 Query - Get Port Names (GPN_NN)

The Name Server shall, if it receives a GPN_NN request, return a list of Port Identifiers and Port Names registered for the specified Node Name. The format of the GPN_NN request CT_IU is shown in table 68. The requestor supplies the Node Name for which associated Port Identifiers and Port Names are sought.

Table 68 – GPN_NN Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Node Name	8

The format of the Accept CT_IU to a GPN_NN request is shown in table 69.

One or more Port Identifiers and Port Names, registered for the specified Node Name, are returned. Each returned Port Identifier and Port Name is preceded by an 8 bit Control field. The format of the Control field is:

a) Bit 7 is set to zero if the Port Identifier and Port Name following the Control field is not the last Port Identifier and Port Name to be returned by the Accept CT_IU; the bit is set to one if the Port Identifier and Port Name following the Control field is the last Port Identifier and Port Name returned by the Accept CT_IU; and

b) Bits 6 to 0 are reserved.

The Port Identifiers and Port Names may be returned in any order and the order may be different for every request even if the same Port Identifiers and Port Names are returned and the requestor is the same.

Table 69 – Accept CT_IU to GPN_NN Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Control (0 r r r r r r r)	1
Port Identifier #1	3
Reserved	4
Port Name #1	8
...	
Control (1 r r r r r r r)	1
Port Identifier #n	3
Reserved	4
Port Name #n	8

5.2.5.17 Query - Get Symbolic Node Name (GSNN_NN)

The Name Server shall, if it receives a GSNN_NN request, return the registered Symbolic Node Name object for the specified Node Name. The format of the GSNN_NN request CT_IU is shown in table 70. The requestor supplies the Node Name for which the Symbolic Node Name is sought.

Table 70 – GSNN_NN Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Node Name	8

The format of Accept CT_IU to a GSNN_NN request is shown in table 71.

The Name length field shall contain a single byte unsigned value indicating the size of the variable length Symbolic Node Name.

The Symbolic Node Name field returns the registered Symbolic Node Name object for the specified Node Name.

Table 71 – Accept CT_IU to GSNN_NN Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Name length (n)	1
Symbolic Node Name	n
Reserved	255-n

5.2.5.18 Query - Get Port Identifiers (GID_FT)

The Name Server shall, if it receives a GID_FT request, return all Port Identifiers having registered support for the specified FC-4 type. The format of the GID_FT request CT_IU is shown in table 72. The requestor supplies the FC-4 type code (see FC-FS-4) for which supporting Port Identifiers are sought.The settings of bits in the Flags field affects operation of the request. The format of the Flags field is as follows:

a) Bit 7 is the Area_ID flag. If the Area_ID flag is set to zero, an Area_ID Scope field value of zero specifies that all area values are in the scope of the request. If the Area_ID flag is set to one, an Area_ID Scope field value of zero specifies that only area zero is in the scope of the request; and

b) Bits 6 to 0 - reserved.

NOTE 17 – The type is specified as an 8-bit encoded FC-FS-4 value, not as an FC-4 type object.

The Domain_ID Scope field, Area_ID Scope field, and the Area_ID flag specify the scope of the request. If both the Domain_ID Scope and the Area_ID Scope fields are zero and the Area_ID flag is set to zero, the Name Server shall return all Port Identifiers having registered support for the specified FC-4 type code. If the Domain_ID Scope field is non-zero, the Area_ID Scope field is zero, and Area_ID flag is set to zero, the Name Server shall return Port Identifiers within the specified Domain having registered support for the specified FC-4 type code. If the Area_ID Scope field is non-zero or the Area_ID flag is set to one, the Name Server shall return Port Identifiers within the specified Domain and Area having registered support for the specified FC-4 type code.

NOTE 18 – Suitable values for the Domain_ID Scope and Area_ID Scope fields may be discovered using the GID_A query.

Table 72 – GID_FT Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Flags	1
Domain_ID Scope	1
Area_ID Scope	1
FC-4 type Code	1

The format of the Accept CT_IU to a GID_FT request is shown in table 73.

One or more Port Identifiers, having registered support for the specified FC-4 type, are returned. Each returned Port Identifier is preceded by an 8 bit Control field. The format of the Control field is:

a) Bit 7 is set to zero if the Port Identifier following the Control field is not the last Port Identifier to be returned by the Accept CT_IU; the bit is set to one if the Port Identifier following the Control field is the last Port Identifier returned by the Accept CT_IU; and

b) Bits 6 to 0 are reserved.

The Port Identifiers may be returned in any order and the order may be different for every request even if the same Port Identifiers are returned and the requestor is the same.

Table 73 – Accept CT_IU to GID_FT Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Control (0 r r r r r r r)	1
Port Identifier #1	3
...	
Control (1 r r r r r r r)	1
Port Identifier #n	3

5.2.5.19 Query - Get Port Names (GPN_FT)

The Name Server shall, if it receives a GPN_FT request, return a list of Port Identifiers and Port Names having registered support for the specified FC-4 type. The format of the GPN_FT request CT_IU is shown in table 74. The requestor supplies the FC-4 type code (see FC-FS-4) for which supporting Port Identifiers and Port Names are sought.The settings of bits in the Flags field affects operation of the request. The format of the Flags field is as follows:

a) Bit 7 is the Area_ID flag. If the Area_ID flag is set to zero, an Area_ID Scope field value of zero specifies that all area values are in the scope of the request. If the Area_ID flag is set to one, an Area_ID Scope field value of zero specifies that only area zero is in the scope of the request; and

b) Bits 6 to 0 - reserved.

NOTE 19 – The type is specified as an 8-bit encoded FC-FS-4 value, not as an FC-4 type object.

The Domain_ID Scope field, Area_ID Scope field, and the Area_ID flag specify the scope of the request. If both the Domain_ID Scope and the Area_ID Scope fields are zero and the Area_ID flag is set to zero, the Name Server shall return all Port Identifiers and Port Names having registered support for the specified FC-4 type code. If the Domain_ID Scope field is non-zero, the Area_ID Scope field is zero, and Area_ID flag is set to zero, the Name Server shall return Port Identifiers and Port Names within the specified Domain having registered support for the specified FC-4 type code. If the Area_ID Scope field is non-zero or the Area_ID flag is set to one, the Name Server shall return Port Identifiers and Port Names within the specified Domain and Area having registered support for the specified FC-4 type code.

NOTE 20 – Suitable values for the Domain_ID Scope and Area_ID Scope fields may be discovered using the GID_A query.

Table 74 – GPN_FT Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Flags	1
Domain_ID Scope	1
Area_ID Scope	1
FC-4 type	1

The format of the Accept CT_IU to a GPN_FT request is shown in table 75.

One or more Port Identifiers and Port Names, having registered support for the specified FC-4 type, are returned. Each returned Port Identifier and Port Name is preceded by an 8 bit Control field. The format of the Control field is:

a) Bit 7 is set to zero if the Port Identifier and Port Name following the Control field is not the last Port Identifier and Port Name to be returned by the Accept CT_IU; the bit is set to one if the Port Identifier and Port Name following the Control field is the last Port Identifier and Port Name returned by the Accept CT_IU; and

b) Bits 6 to 0 are reserved.

The Port Identifiers and Port Names may be returned in any order and the order may be different for every request even if the same Port Identifiers and Port Names are returned and the requestor is the same.

Table 75 – Accept CT_IU to GPN_FT Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Control (0 r r r r r r r)	1
Port Identifier #1	3
Reserved	4
Port Name #1	8
...	
Control (1 r r r r r r r)	1
Port Identifier #n	3
Reserved	4
Port Name #n	8

5.2.5.20 Query - Get Node Names (GNN_FT)

The Name Server shall, if it receives a GNN_FT request, return a list of Port Identifiers and Node Names having registered support for the specified FC-4 type. The format of the GNN_FT request CT_IU is shown in table 76. The requestor supplies the FC-4 type code (see FC-FS-4) for which supporting Port Identifiers and Node Names are sought. The settings of bits in the Flags field affects operation of the request. The format of the Flags field is as follows:

a) Bit 7 is the Area_ID flag. If the Area_ID flag is set to zero, an Area_ID Scope field value of zero specifies that all area values are in the scope of the request. If the Area_ID flag is set to one, an Area_ID Scope field value of zero specifies that only area zero is in the scope of the request; and

b) Bits 6 to 0 - reserved.

NOTE 21 – The type is specified as an 8-bit encoded FC-FS-4 value, not as an FC-4 type object.

The Domain_ID Scope field, Area_ID Scope field, and the Area_ID flag specify the scope of the request. If both the Domain_ID Scope and the Area_ID Scope fields are zero and the Area_ID flag is set to zero, the Name Server shall return all Port Identifiers and Node Names having registered support for the specified FC-4 type code. If the Domain_ID Scope field is non-zero, the Area_ID

Scope field is zero, and Area_ID flag is set to zero, the Name Server shall return Port Identifiers and Node Names within the specified Domain having registered support for the specified FC-4 type code. If the Area_ID Scope field is non-zero or the Area_ID flag is set to one, the Name Server shall return Port Identifiers and Node Names within the specified Domain and Area having registered support for the specified FC-4 type code.

NOTE 22 – Suitable values for the Domain_ID Scope and Area_ID Scope fields may be discovered using the GID_A query.

Table 76 – GNN_FT Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Flags	1
Domain_ID Scope	1
Area_ID Scope	1
FC-4 type	1

The format of the Accept CT_IU to a GNN_FT request is shown in table 77.

One or more Port Identifiers and Node Names, having registered support for the specified FC-4 type, are returned. Each returned Port Identifier and Node Name is preceded by an 8 bit Control field. The format of the Control field is:

a) Bit 7 is set to zero if the Port Identifier and Node Name following the Control field is not the last Port Identifier and Node Name to be returned by the Accept CT_IU; the bit is set to one if the Port Identifier and Node Name following the Control field is the last Port Identifier and Node Name returned by the Accept CT_IU; and

b) Bits 6 to 0 are reserved.

The Port Identifiers and Node Names may be returned in any order and the order may be different for every request even if the same Port Identifiers and Node Names are returned and the requestor is the same.

Table 77 – Accept CT_IU to GNN_FT Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Control (0 r r r r r r r)	1
Port Identifier #1	3
Reserved	4
Node Name #1	8
...	
Control (1 r r r r r r r)	1
Port Identifier #n	3
Reserved	4
Node Name #n	8

5.2.5.21 Query - Get Node Names (GNN_FF)

The Name Server shall, if it receives a GNN_FF request, return all Port Identifiers and Node Names having registered support for the specified type code value and corresponding FC-4 features. The format of the GNN_FF request CT_IU is shown in table 78. The requestor supplies the type code and FC-4 Feature Bits for which supporting Port Identifiers and Node Names are sought. The format of the FC-4 Feature Bits field is:

a) Bits 7 to 4 are reserved; and

b) Bits 3 to 0 contain the four FC-4 Feature Bits for the specified type code.

The settings of bits in the Flags field affects operation of the request. The format of the Flags field is as follows:

a) Bit 7 is the Area_ID flag. If the Area_ID flag is set to zero, an Area_ID Scope field value of zero specifies that all area values are in the scope of the request. If the Area_ID flag is set to one, an Area_ID Scope field value of zero specifies that only area zero is in the scope of the request; and

b) Bits 6 to 0 ~~–~~reserved.

The Domain_ID Scope field, Area_ID Scope field, and the Area_ID flag specify the scope of the request. If both the Domain_ID Scope and the Area_ID Scope fields are zero and the Area_ID flag is set to zero, the Name Server shall return all Port Identifiers and Node Names having registered for the specified type code and FC-4 Feature Bits. If the Domain_ID Scope field is non-zero, the Area_ID Scope field is zero, and Area_ID flag is set to zero, the Name Server shall return Port Identifiers and Node Names within the specified Domain having registered for the specified type code and FC-4 Feature Bits. If the Area_ID Scope field is non-zero or the Area_ID flag is set to one, the Name Server shall return Port Identifiers and Node Names within the specified Domain and Area having registered for the specified type code and FC-4 Feature Bits.

NOTE 23 – Suitable values for the Domain_ID Scope and Area_ID Scope fields may be discovered using the GID_A query.

Table 78 – GNN_FF Request CT_IU

Item	Size (Bytes)
CT_UI preamble	see 4.3
Flags	1
Domain_ID scope	1
Area_ID scope	1
Reserved	1
Reserved	2
FC-4 Feature Bits	1
Type code	1

The format of the Accept CT_IU to a GPN_FT request is shown in table 79.

One or more Port Identifiers and Node Names, having registered the specified FC-4 features, are returned. Each returned Port Identifier and Node Name is preceded by an 8 bit Control field. The format of the Control field is:

a) Bit 7 is set to zero if the Port Identifier following the Control field is not the last Port Identifier to be returned by the Accept CT_IU; the bit is set to one if the Port Identifier following the Control field is the last Port Identifier and Node Name returned by the Accept CT_IU; and

b) Bits 6 to 0 are reserved.

The Port Identifiers and Node Names may be returned in any order and the order may be different for every request even if the same Port Identifiers and Node Names are returned and the requestor is the same.

Table 79 – GNN_FF Accept CT_IU (Same as GNN_FT)

Item	Size (Bytes)
CT_IU preamble	see 4.3
Control (0 r r r r r r r)	1
Port Identifier #1	3
Reserved	4
Node Name #1	8
...	
Control (1 r r r r r r r)	1
Port Identifier #n	3
Reserved	4
Node Name #n	8

5.2.5.22 Query - Get Port Names (GPN_FF)

The Name Server shall, if it receives a GPN_FF request, return all Port Identifiers and Port Names having registered support for the specified type code value and corresponding FC-4 Feature Bits. The format of the GPN_FF request CT_IU is shown in table 80. The requestor supplies the type code and FC-4 Feature Bits for which supporting Port Identifiers and Port Names are sought. The format of the FC-4 Feature Bits field is:

a) Bits 7 to 4 are reserved; and

b) Bits 3 to 0 contain the four FC-4 Feature Bits for the specified type code.

The settings of bits in the Flags field affects operation of the request. The format of the Flags field is as follows:

a) Bit 7 is the Area_ID flag. If the Area_ID flag is set to zero, an Area_ID Scope field value of zero specifies that all area values are in the scope of the request. If the Area_ID flag is set to one, an Area_ID Scope field value of zero specifies that only area zero is in the scope of the request; and

b) Bits 6 to 0 ~~–~~ reserved.

The Domain_ID Scope field, Area_ID Scope field, and the Area_ID flag specify the scope of the request. If both the Domain_ID Scope and the Area_ID Scope fields are zero and the Area_ID flag is set to zero, the Name Server shall return all Port Identifiers and Port Names having registered for the specified type code and FC-4 Feature Bits. If the Domain_ID Scope field is non-zero, the Area_ID

Scope field is zero, and Area_ID flag is set to zero, the Name Server shall return Port Identifiers and Port Names within the specified Domain having registered for the specified type code and FC-4 Feature Bits. If the Area_ID Scope field is non-zero or the Area_ID flag is set to one, the Name Server shall return Port Identifiers and Port Names within the specified Domain and Area having registered for the specified type code and FC-4 Feature Bits.

NOTE 24 – Suitable values for the Domain_ID Scope and Area_ID Scope fields may be discovered using the GID_A query.

Table 80 – GPN_FF Request CT_IU

Item	Size (Bytes)
CT_UI preamble	see 4.3
Flags	1
Domain_ID scope	1
Area_ID scope	1
Reserved	1
Reserved	2
FC-4 Feature Bits	1
Type code	1

The format of the Accept CT_IU to a GPN_FF request is shown in table 81.

One or more Port Identifiers and Port Names, having registered support for the specified FC-4 features, are returned. Each returned Port Identifier and Port Name is preceded by an 8 bit Control field. The format of the Control field is:

a) Bit 7 is set to zero if the Port Identifier and Port Name following the Control field is not the last Port Identifier and Port Name to be returned by the Accept CT_IU; the bit is set to one if the Port Identifier and Port Name following the Control field is the last Port Identifier and Port Name returned by the Accept CT_IU; and

b) Bits 6 to 0 are reserved.

The Port Identifiers and Node Names may be returned in any order and the order may be different for every request even if the same Port Identifiers and Node Names are returned and the requestor is the same.

Table 81 – GPN_FF Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Control (0 r r r r r r r)	1
Port Identifier #1	3
Reserved	4
Port Name #1	8
...	
Control (1 r r r r r r r)	1
Port Identifier #n	3
Reserved	4
Port Name #n	8

5.2.5.23 Query - Get Port Names - Simplified Discovery for FCP (GPN_SDFCP)

The Name Server shall, if it receives a GPN_SDFCP request, return Port Identifiers and Port Names for all Nx_Ports having registered support for the FCP type code (i.e., 08h) and FC-4 Feature Bits that match the combination specified by the FCP Feature selection bits (see table 84). The format of the GPN_SDFCP request CT_IU is shown in table 82.

Table 82 – GPN_SDFCP Request CT_IU

Item	Size (Bytes)
CT_UI preamble	see 4.3
Flags	1
Domain_ID scope	1
Area_ID scope	1
Reserved	1
Reserved	3
FC-4 Feature Selection Bits	1

The settings of bits in the Flags field affects operation of the request. The format of the Flags field is as follows:

a) Bit 7 is the Area_ID flag. If the Area_ID flag is set to zero, an Area_ID Scope field value of zero specifies that all area values are in the scope of the request. If the Area_ID flag is set to one, an Area_ID Scope field value of zero specifies that only area zero is in the scope of the request; and

b) Bits 6 to 0 reserved.

As shown in table 83, the Domain_ID Scope and Area_ID Scope fields specify the scope of the request.

Table 83 – Domain and Area scope

Domain ID Scope	Area ID scope	Flags (bit 7)	Port Identifiers returned
0	0	0	All Port Identifiers and Port Names having registered the FCP type (i.e., 08h) and the specified FC-4 Feature Bits, see table 84.
Non-zero	0	0	Port Identifiers and Port Names within the specified Domain having registered the FCP type code (i.e., 08h) and FC-4 Feature Bits, see table 84.
Non-zero	0	1	Port Identifiers and Port Names within the specified Domain and an Area of 00h having registered the FCP type code (i.e., 08h) and FC-4 Feature Bits, see table 84.
Non-zero	Non-zero	N/A	Port Identifiers and Port Names within the specified Domain and Area having registered the FCP type code (i.e., 08h) and FC-4 Feature Bits, see table 84.

NOTE 25 – Suitable values for the Domain_ID Scope and Area_ID Scope fields may be discovered using the GID_A query.

The FC-4 Feature Selection Bits field shall contain the feature(s) being requested as follows:

a) initiator function selection (bit 1); and

b) target function selection (bit 0).

FC-4 Feature selection bit values in the request other than the initiator function selection bit and target function selection bit shall be reserved.

The initiator function bit and target function bit in table 84 shall be as defined in FCP-4.

The FC-4 Feature selection field usage is shown in table 84.

Table 84 – Feature selection bit usage

Initiator function selection bit value in request	Target function selection bit value in request	Selection criteria for returned Nx_Ports
0	0	No Nx_Ports are returned.
0	1	Returns FCP Nx_Ports with either target function bit set to one, or all feature bits set to zero.
1	0	Returns FCP Nx_Ports with initiator function bit set to one, or all feature bits set to zero.
1	1	All FCP Nx_Ports are returned.

The format of the Accept CT_IU to the GPN_SDFCP request is shown in table 85.

Table 85 – Accept CT_IU to GPN_SDFCP Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Control (0 r r r r r r r)	1
Port Identifier #1	3
Port Name #1	8
FCP Feature bits	1
Additional FCP Features bits	1
Generic Fibre Channel features bits	1
Reserved	1
...	
Control (1 r r r r r r r)	1
Port Identifier #n	3
Port Name #n	8
FCP Feature bits	1
Additional FCP Features bits	1
Generic Fibre Channel features bits	1
Reserved	1

One or more Port Identifiers and Port Names, having registered support for the specified FC-4 features, are returned.

The Control field (i.e., bit 7) is set to zero if the Port Identifier and Port Name following the Control field is not the last Port Identifier and Port Name to be returned by the Accept CT_IU. The Control field is set to one if the Port Identifier and Port Name following the Control field is the last Port Identifier and Port Name returned by the Accept CT_IU. Bits 6 to 0 are reserved.

Bits 3 to 0 of the FCP Feature field shall contain the four FCP FC-4 Feature Bits (see FCP-4) registered by the Nx_Port. Bits 7 to 4 are reserved.

Bits 3 to 0 of the Additional FCP Features field shall contain the four Additional FCP Features bits (see FCP-4) registered by the Nx_Port, if any. Bits 7 to 4 are reserved.

Bits 3 to 0 of the Generic Fibre Channel Feature shall contain the four Generic Fibre Channel Features bits (see Annex A) registered by the Nx_Port, if any. Bits 7 to 4 are reserved.

The Port Identifiers and Node Names may be returned in any order and the order may be different for every request even if the same Port Identifiers and Node Names are returned and the requestor is the same.

5.2.5.24 Query - Get Port Identifiers (GID_PT)

The Name Server shall, if it receives a GID_PT request, return all Port Identifiers having registered support for the specified Port Type. If the specified Port Type is equal to ‘Nx_Port’, then the Name Server shall return all Port Identifiers that have registered Port Types with an unsigned value of less than 80h (e.g., Port Identifiers for all registered unidentified ports, N_Ports, NL_Ports, F/NL_Ports). The format of the GID_PT request CT_IU is shown in table 86. The requestor supplies the Port Type for which supporting Port Identifiers are sought. The settings of bits in the Flags field affects operation of the request. The format of the Flags field is as follows:

a) Bit 7 is the Area_ID flag. If the Area_ID flag is set to zero, an Area_ID Scope field value of zero specifies that all area values are in the scope of the request. If the Area_ID flag is set to one, an Area_ID Scope field value of zero specifies that only area zero is in the scope of the request; and

b) Bits 6 to 0 ~~-~~reserved.

The Domain_ID Scope field, Area_ID Scope field, and the Area_ID flag specify the scope of the request. If both the Domain_ID Scope and the Area_ID Scope fields are zero and the Area_ID flag is set to zero, the Name Server shall return all Port Identifiers having registered support for the specified Port Type. If the Domain_ID Scope field is non-zero, the Area_ID Scope field is zero, and Area_ID flag is set to zero, the Name Server shall return Port Identifiers within the specified Domain having registered support for the specified Port Type. If the Area_ID Scope field is non-zero or the

Area_ID flag is set to one, the Name Server shall return Port Identifiers within the specified Domain and Area having registered support for the specified Port Type.

NOTE 26 – Suitable values for the Domain_ID Scope and Area_ID Scope fields may be discovered using the GID_A query.

Table 86 – GID_PT Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Port Type	1
Domain_ID Scope	1
Area_ID Scope	1
Flags	1

The format of the Accept CT_IU to a GID_PT request is shown in table 87.

One or more Port Identifiers, having registered as the specified Port Type, are returned. Each returned Port Identifier is preceded by an 8 bit Control field. The format of the Control field is:

a) Bit 7 is set to zero if the Port Identifier following the Control field is not the last Port Identifier to be returned by the Accept CT_IU; the bit is set to one if the Port Identifier following the Control field is the last Port Identifier returned by the Accept CT_IU; and

b) Bits 6 to 0 are reserved.

The Port Identifiers may be returned in any order and the order may be different for every request even if the same Port Identifiers are returned and the requestor is the same.

Table 87 – Accept CT_IU to GID_PT Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Control (0 r r r r r r r)	1
Port Identifier #1	3
...	
Control (1 r r r r r r r)	1
Port Identifier #n	3

5.2.5.25 Query - Get Port Identifiers - Fabric Port Name (GID_FPN)

The Name Server shall, if it receives a GID_FPN request, return all Port Identifiers registered for the specified Fabric Port Name. The format of the GID_FPN request CT_IU is shown in table 88. The requestor supplies the Fabric Port Name for which associated Port Identifiers are sought.

Table 88 – GID_FPN Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Fabric Port Name	8

The format of the Accept CT_IU to a GID_FPN request is shown in table 89.

One or more Port Identifiers are returned. Each returned Port Identifier is preceded by an 8 bit Control field. The format of the Control field is:

a) Bit 7 is set to zero if the Port Identifier following the Control field is not the last Port Identifier to be returned by the Accept CT_IU; the bit is set to one if the Port Identifier following the Control field is the last Port Identifier returned by the Accept CT_IU; and

b) Bits 6 to 0 are reserved.

Table 89 – Accept CT_IU to GID_FPN Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Control (0 r r r r r r r)	1
Port Identifier #1	3
...	
Control (1 r r r r r r r)	1
Port Identifier #n	3

A single entry with a null Port Identifier with Control Bit 7 set to one indicates that there are no matching entries. A single Port Identifier is returned if there is a single Nx_Port device attached. If there are multiple NL_Port devices attached, the Port Identifiers may be returned in any order, and the order may be different for every request even if the same Port Identifiers are returned and the requestor is the same.

5.2.5.26 Query - Get Permanent Port Name - Port Identifier (GPPN_ID)

The Name Server shall, if it receives a GPPN_ID request, return the Permanent Port Name object for the specified Port Identifier. The format of the GPPN_ID request CT_IU is shown in table 90. The requestor supplies the Port Identifier for which the Permanent Port Name object is sought.

Table 90 – GPPN_ID Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Reserved	1
Port Identifier	3

The format of the Accept CT_IU to a GPPN_ID request is shown in table 91.

The Permanent Port Name field returns the registered value for the Permanent Port Name.

Table 91 – Accept CT_IU to GPPN_ID Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Permanent Port Name	8

5.2.5.27 Query - Get Port Identifiers (GID_FF)

The Name Server shall, if it receives a GID_FF request, return all Port Identifiers having registered support for the specified type code value and corresponding FC-4 Feature Bits. The format of the GID_FF request CT_IU is shown in table 92. The requestor supplies the type code and FC-4 Feature Bits for which supporting Port Identifiers are sought. The format of the FC-4 Feature Bits field is:

a) Bits 7 to 4 are reserved; and

b) Bits 3 to 0 contain the four FC-4 Feature Bits for the specified type code.

The settings of bits in the Flags field affects operation of the request. The format of the Flags field is as follows:

a) Bit 7 is the Area_ID flag. If the Area_ID flag is set to zero, an Area_ID Scope field value of zero specifies that all area values are in the scope of the request. If the Area_ID flag is set to one, an Area_ID Scope field value of zero specifies that only area zero is in the scope of the request; and

b) Bits 6 to 0 ~~-~~ reserved.

The Domain_ID Scope field, Area_ID Scope field, and the Area_ID flag specify the scope of the request. If both the Domain_ID Scope and the Area_ID Scope fields are zero and the Area_ID flag is set to zero, the Name Server shall return all Port Identifiers having registered for the specified type code and FC-4 Feature Bits. If the Domain_ID Scope field is non-zero, the Area_ID Scope field is zero, and Area_ID flag is set to zero, the Name Server shall return Port Identifiers within the specified Domain having registered for the specified type code and FC-4 Feature Bits. If the Area_ID Scope field is non-zero or the Area_ID flag is set to one, the Name Server shall return Port Identifiers within the specified Domain and Area having registered for the specified type code and FC-4 Feature Bits.

NOTE 27 – Suitable values for the Domain_ID Scope and Area_ID Scope fields may be discovered using the GID_A query.

Table 92 – GID_FF Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Flags	1
Domain_ID Scope	1
Area_ID Scope	1
Reserved	1
Reserved	2
FC-4 Feature Bits	1
Type code	1

The format of the Accept CT_IU to a GID_FF request is shown in table 93.

One or more Port Identifiers, having registered the specified FC-4 Features, are returned. Each returned Port Identifier is preceded by an 8 bit Control field. The format of the Control field is:

a) Bit 7 is set to zero if the Port Identifier following the Control field is not the last Port Identifier to be returned by the Accept CT_IU; the bit is set to one if the Port Identifier following the Control field is the last Port Identifier returned by the Accept CT_IU; and

b) Bits 6 to 0 are reserved.

The Port Identifiers may be returned in any order and the order may be different for every request even if the same Port Identifiers are returned and the requestor is the same.

Table 93 – Accept CT_IU to GID_FF Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Control (0 r r r r r r r)	1
Port Identifier #1	3
...	
Control (1 r r r r r r r)	1
Port Identifier #n	3

5.2.5.28 Query - Get Port Identifier (Domain/Port) (GID_DP)

The Name Server shall, if it receives a GID_DP request, return the Port Identifiers associated with the specified Domain_ID and Physical Port Number. The format of the GID_DP request CT_IU is shown in table 94. The requestor supplies the Domain_ID and Physical Port Number for which the Port Identifiers are sought.

Table 94 – GID_DP Request CT_IU

Item	Size (Bytes)
CT_IU Preamble	see 4.3
Domain_ID	1
Physical Port Number	2
reserved	1

The format of the Accept CT_IU to a GID_DP request is shown in table 95. One or more Port Identifiers are returned. Each returned Port Identifier is preceded by an 8 bit Control field. The format of the Control field is:

a) Bit 7 is set to zero if the Port Identifier following the Control field is not the last Port Identifier to be returned by the Accept CT_IU; the bit is set to one if the Port Identifier following the Control field is the last Port Identifier returned by the Accept CT_IU; and

b) Bits 6 to 0 are reserved.

The Port Identifier field returns the registered Port Identifier value for the specified Domain_ID and Physical Port Number. The Port Identifiers may be returned in any order and the order may be different for every request even if the same Port Identifiers are returned and the requestor is the same.

Table 95 – Accept CT_IU to GID_DP Request

Item	Size (Bytes)
CT_IU Preamble	see 4.3
Control (0 r r r r r r r)	1
Port Identifier #1	3
...	
Control (1 r r r r r r r)	1
Port Identifier #n	3

5.2.5.29 Register Node Name (RNN_ID)

The RNN_ID Name Server request shall be used to associate a Node Name with a given Port Identifier.

The Fabric may register the Node Name for a Port Identifier before explicit Fabric Login (FLOGI) has completed.

The Name Server shall not attempt validation of the Node Name object and shall accept any 64 bit value.

Deregistration may be accomplished by registering a null Node Name (see 5.2.3.4).

The format of the RNN_ID request CT_IU is shown in table 96.

Table 96 – RNN_ID Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Reserved	1
Port Identifier	3
Node Name	8

The format of the RNN_ID Accept CT_IU is shown in table 97.

Table 97 – RNN_ID Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

5.2.5.30 Register Class of Service (RCS_ID)

The RCS_ID Name Server request shall be used to record which Classes of Service are supported by a given Port Identifier.

The Fabric may register the Class of Service for a Port Identifier before explicit Fabric Login (FLOGI) has completed.

The Name Server shall not attempt validation of the Class of Service object and shall accept any 32 bit value.

Deregistration may be accomplished by registering a null Class of Service object (see 5.2.3.5).

The format of the RCS_ID request CT_IU is shown in table 98.

Table 98 – RCS_ID Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Reserved	1
Port Identifier	3
Class of Service	4

The format of the RCS_ID Accept CT_IU is shown in table 99.

Table 99 – RCS_ID Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

5.2.5.31 Register FC-4 types (RFT_ID)

The RFT_ID Name Server request shall be used to record which FC-4 types are supported by a given Port Identifier.

The Name Server shall not attempt validation of the FC-4 types object and shall accept any 32 byte value.

Deregistration may be accomplished by registering a null FC-4 types object (see 5.2.3.6).

The format of the RFT_ID request CT_IU is shown in table 100.

Table 100 – RFT_ID Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Reserved	1
Port Identifier	3
FC-4 types	32

The format of the RFT_ID Accept CT_IU is shown in table 101.

Table 101 – RFT_ID Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

5.2.5.32 Register Symbolic Port Name (RSPN_ID)

The RSPN_ID Name Server request shall be used to associate a Symbolic Port Name with a given Port Identifier.

The Name Server shall not attempt validation of the Symbolic Port Name object and shall accept any value up to 255 bytes in length.

Deregistration may be accomplished by registering a null Symbolic Port Name object (see 5.2.3.7).

The format of the RSPN_ID request CT_IU is shown in table 102. The Name length field shall contain a single byte unsigned value indicating the size of the variable length Symbolic Port Name.

Table 102 – RSPN_ID Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Reserved	1
Port Identifier	3
Name length (n)	1
Symbolic Port Name	n

The format of the RSPN_ID Accept CT_IU is shown in table 103.

Table 103 – RSPN_ID Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

5.2.5.33 Register Hard Address (RHA_ID)

The RHA_ID Name Server request shall be used to associate a Hard Address with a given Port Identifier.

The Fabric may register the Hard Address for a Port Identifier before explicit Fabric Login (FLOGI) has completed.

The Name Server shall not attempt validation of the Hard Address object and shall accept any 3 byte value.

Deregistration may be accomplished by registering a null Hard Address (see 5.2.3.11).

The format of the RHA_ID request CT_IU is shown in table 104.

Table 104 – RHA_ID Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Reserved	1
Port Identifier	3
Reserved	1
Hard Address	3

The format of the RHA_ID Accept CT_IU is shown in table 105.

Table 105 – RHA_ID Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

5.2.5.34 Register FC-4 Features (RFF_ID)

The RFF_ID Name Server request shall be used to record the FC-4 Features for a specified type that are supported by a given Port Identifier.

If the FC-4 type corresponding to the type code specified in the RFF_ID request CT_IU is not registered, the Name Server shall reject the registration request and shall set the reason code to ‘Unable to perform command request’ and the reason code explanation to ‘FC-4 types not registered’.

The Name Server shall not attempt validation of the FC-4 Features object.

Deregistration may be accomplished by registering a value of zero for the FC-4 Feature Bits.

The format of the RFF_ID request CT_IU is shown in table 106.

Table 106 – RFF_ID Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Reserved	1
Port Identifier	3
Reserved	2
FC-4 Feature Bits	1
Type code	1

The requestor supplies the type code and FC-4 Feature Bits to be registered. The format of the FC-4 Feature Bits field is:

a) Bits 7 to 4 are reserved; and

b) Bits 3 to 0 contain the four FC-4 Feature Bits for the specified type code.

The format of the RFF_ID Accept CT_IU is shown in table 107.

Table 107 – RFF_ID Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

5.2.5.35 Register Symbolic Node Name (RSNN_NN)

The RSNN_NN Name Server request shall be used to associate a Symbolic Node Name with a given Node Name.

Attempts at registration of a Symbolic Node Name shall be rejected by the Name Server unless Node Name registration has been successfully completed (see 5.2.5.29). The Name Server may reject registration of the Symbolic Node Name unless the registration is attempted by one of the Port Identifier associated with the Node Name in the request CT_IU payload.

The Name Server shall not attempt validation of the Symbolic Node Name object and shall accept any value up to 255 bytes in length.

Deregistration may be accomplished by registering a null Symbolic Node Name object (see 5.2.3.8).

The format of the RSNN_NN request CT_IU is shown in table 108. The Name length field shall contain a single byte unsigned value indicating the size of the variable length Symbolic Node Name.

Table 108 – RSNN_NN Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Node Name	8
Name length (n)	1
Symbolic Node Name	n

The format of the RSNN_NN Accept CT_IU is shown in table 109.

Table 109 – RSNN_NN Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

If the RSNN_NN Name Server request is rejected by the Name Server because the Node Name is not registered with the Name Server, then the Reject CT_IU reason code shall be ‘Unable to perform command request’, with a reason code explanation of ‘Node Name not registered’;

5.2.5.36 Remove all (DA_ID)

The DA_ID shall be used to delete all entries and associations for a given Port Identifier in the Name Server’s data base.

The Name Server shall process DA_ID requests received from the Nx_Port with its address identifier equal to the Port Identifier in the request CT_IU payload, from the Link Service Facilitator and from the Fabric Controller. The Name Server shall reject a DA_ID request from any other source.

The Fabric should not issue the DA_ID Name Server request, unless the address identifier is removed as a Port Identifier, has disappeared from the Fabric or if the address identifier has been reused.

The format of the DA_ID request CT_IU is shown in table 110. The Port Identifier format shall be as defined in 5.2.3.2.

Table 110 – DA_ID Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Reserved	1
Port Identifier	3

The format of the DA_ID Accept CT_IU is shown in table 111.

Table 111 – DA_ID Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

If the DA_ID Name Server request is rejected by the Name Server because the Port Identifier is not registered with the Name Server, then the Reject CT_IU reason code shall be ‘Unable to perform command request’, with a reason code explanation of ‘Port Identifier not registered’.

5.3 VE Identification Server

5.3.1 Overview

The VE Identification Server maintains the mappings between Global VE IDs and Fabric VE IDs used by the Priority Tagging mechanism (see FC-LS-3). Table 112 shows the mappings maintained by the VE Identification Server.

Table 112 – VE Identification Server Mappings

Global VE ID	Fabric VE ID	
Global VE ID #1	N_Port_ID #1	Local VE ID #1
Global VE ID #2	N_Port_ID #2	Local VE ID #2
...		
Global VE ID #n	N_Port_ID #n	Local VE ID #n

The VE Identification Server shall allow a Global VE ID to be mapped to more than one Fabric VE ID.

NOTE 28 – A Global VE ID is mapped to more than one Fabric VE ID to enable scenarios in which a VE is associated with more than one Fabric VE ID to provide to that VE different QoS priorities, or corner cases of VE movements, where a VE is instantiated on the destination VEM before being deinstantiated on the origin VEM.

5.3.2 Protocol

Registrations and deregistrations in the VE Identification Server are managed using the UVEM ELS (see FC-LS-3). Queries are defined to return information from the VE Identification Server.

For a VE Identification Server request, the request payload shall be transported from the requestor to the VE Identification Server using a request CT_IU. The corresponding VE Identification Server response is transported from the VE Identification Server to the requestor, in the Exchange established by the requestor, using a response CT_IU. The following values shall be set in the CT_IU preamble for VE Identification Server requests and their responses; fields not specified here shall be set as defined in 4.3.2:

a) GS_Subtype: as indicated in table 19; and

b) Command Code: see table 113 for request command codes.

Table 113 – VE Identification Server - Request Command Codes

Code	Mnemonic	Description	Reference subclause
0100h	G_GVID	Get Global VE ID	5.3.3.2
0101h	G_FVID	Get Fabric VE ID	5.3.3.3
0102h	G_VEMID	Get VEM IDs	5.3.3.4
0103h	G_VEM	Get VE Mappings	5.3.3.5

5.3.3 Commands

5.3.3.1 Overview

The commands defined for the VE Identification Server are summarized in table 113.

5.3.3.2 Query - Get Global VE ID (G_GVID)

The VE Identification Server shall, when it receives a G_GVID request, return the Global VE ID(s) associated with the specified Fabric VE ID. The format of the G_GVID Request CT_IU is shown in table 114.

Table 114 – G_GVID Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Fabric VE ID	see table 115

The Fabric VE ID format is shown in table 115.

Table 115 – Fabric VE ID Format

Item	Size (Bytes)
N_Port_ID	3
Local VE ID	1

N_Port_ID: the N_Port_ID of the VE.

Local VE ID: the 8-bit local VE ID of the VE. The value 00h is used to request all global VE IDs associated with the specified N_Port_ID.

The format of the Accept CT_IU to a G_GVID request is shown in table 116.

Table 116 – Accept CT_IU to G_GVID Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Number of Global VE IDs (s)	4
Global VE ID #1	16
Global VE ID #2	16
...	
Global VE ID #s	16

Global VE ID: contains the global VE ID of the identified VE(s).

5.3.3.3 Query - Get Fabric VE ID (G_FVID)

The VE Identification Server shall, when it receives a G_FVID request, return the Fabric VE ID(s) associated with the specified global VE ID. The format of the G_FVID Request CT_IU is shown in table 117.

Table 117 – G_FVID Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Global VE ID	16

Global VE ID: contains the global VE ID of the VE whose Fabric VE ID(s) is being requested.

The format of the Accept CT_IU to a G_FVID request is shown in table 118.

Table 118 – Accept CT_IU to G_FVID Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Number of Fabric VE IDs (t)	4
Fabric VE ID #1	see table 115
Fabric VE ID #2	see table 115
...	
Fabric VE ID #t	see table 115

Fabric VE ID: contains the Fabric VE ID(s) of the identified VE. The Fabric VE ID format is shown in table 115.

5.3.3.4 Query - Get VEM IDs (G_VEMID)

The VE Identification Server shall, when it receives a G_VEMID request, return the list of VEM IDs registered in the VE Identification Server along with their registered N_Port_IDs. The format of the G_VEMID Request CT_IU is shown in table 119.

Table 119 – G_VEMID Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

The format of the Accept CT_IU to a G_VEMID request is shown in table 120.

Table 120 – Accept CT_IU to G_VEMID Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Number of VEM Records (u)	4
VEM Record #1	see table 121
VEM Record #2	see table 121
...	
VEM Record #u	see table 121

The format of the VEM record is shown in table 121.

Table 121 – VEM Record Format

Item	Size (Bytes)
VEM ID	16
Number of N_Port_IDs (x)	4
N_Port_ID #1	4
N_Port_ID #2	4
...	
N_Port_ID #x	4

VEM ID: contains a VEM ID registered in the VE Identification Server.

N_Port_ID: contains in the lowest significant three bytes an N_Port_ID associated with the described VEM. The most significant byte is reserved.

5.3.3.5 Query - Get VE Mappings (G_VEM)

The VE Identification Server shall, when it receives a G_VEM request, return the VE Mappings associated with the specified VEM ID and N_Port_ID. The format of the G_VEM Request CT_IU is shown in table 122.

Table 122 – G_VEM Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
VEM ID	16
N_Port_ID	4

VEM ID: contains the VEM ID of the specified VEM.

N_Port_ID: contains in the lowest significant three bytes the N_Port_ID associated with the specified VEM whose VE Mappings are being requested. The most significant byte is reserved.

The format of the Accept CT_IU to a G_VEM request is shown in table 123.

Table 123 – Accept CT_IU to G_VEM Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Number of VE Mapping Records (v)	4
VE Mapping Record #1	see table 124
VE Mapping Record #2	see table 124
...	
VE Mapping Record #v	see table 124

The format of the VE Mapping record is shown in table 124.

Table 124 – VE Mapping Record Format

Item	Size (Bytes)
Global VE ID	16
Number of Fabric VE IDs (t)	4
Fabric VE ID #1	see table 115
Fabric VE ID #2	see table 115
...	
Fabric VE ID #t	see table 115

Global VE ID: contains the global VE ID of the described VE Mapping.

Fabric VE ID: contains the Fabric VE ID(s) of the described VE Mapping. The Fabric VE ID format is shown in table 115.

6 Management Service

6.1 Overview

The Management Service provides a single management access point within the Fibre Channel Fabric. Management Service covers the following areas:

a) Fabric Configuration Server - provides for the configuration management of the Fabric;

b) Unzoned Name Server - provides access to Name Server (see 5.2) information that is not subject to Zone constraints;

c) Fabric Zone Server - provides access to, and control of, Zoning;

d) Performance Server - provides performance metrics for Fabrics;

e) Security Policy Server - provides distribution of security policies;

f) Security Information Server - provides access to operational security information;

g) Fabric Device Management Interface - provides access to data associated with attached devices;

h) Enhanced Fabric Configuration Server - provides for the configuration management of the Fabric; and

i) Application Server - provides for application specific management (e.g. application identifier management).

This standard defines the model for requests and responses to access Management Service information. This standard does not define the structure of this information.

NOTE 29 – Use of CT Authentication (see 4.8) or of other methods to ensure message integrity and authentication (see FC-SP-2) is recommended for use within the Management Service.

The GS_Type for all Management Services shall be set as indicated in Table 4. Table 125 defines the GS_Subtype codes for the Management Service.

Table 125 – Management Service subtype values

Values	Description	Defining Clause
01h	Fabric Configuration Server	see 6.2
02h	Unzoned Name Server	see 6.3
03h	Fabric Zone Server	see 6.4
04h	Reserved for Lock Server	-
05h	Reserved for Performance Server	-
06h	Security Policy Server	see 6.5
07h	Security Information Server	see 6.6
08h	Enhanced Fabric Configuration Server	see 6.8
10h to 1Fh	Fabric Device Management Interface	see 6.7
20h	Application Server	see 6.9
E0h to FFh	Vendor Specific Servers	
other values	Reserved	

The consumer of a Management Service is normally a management application. The Management Service provides for both monitoring and control of the system by the management application. Because Directory Service information is not normally forwarded to an application level (see clause 5), the Management Service provides access to that information via its own services, for use by the management application. In order to provide a system-wide view of management information, the requests of some Servers of the Management Service shall not be constrained by Zoning, as specified in table 126, therefore, some form of authentication and/or access control is desirable.

Table 126 – Zoning effect on Servers of the Management Service

Server	Constrained by Zoning
Fabric Configuration Server	No
Unzoned Name Server	No
Fabric Zone Server	No
Lock Server	No
Performance Server	No
Security Policy Server	No
Fabric Device Management Interface	No
Application Server	No
Security Information Server[a]	-
Vendor Specific Servers[b]	-

[a] The impact of Zoning on the Security Information Server is specified for each Security Information Server request.

[b] The impact of Zoning on Vendor Specific Servers is vendor specific.

6.2 Fabric Configuration Server

6.2.1 Overview

The Fabric Configuration Server provides a way for a management application to discover Fibre Channel Fabric topology and attributes.

Requests for the Fabric Configuration Server are carried over the Common Transport (see clause 4).

The Fabric Configuration Server is intended to be distributed among Switches, making the Fabric Configuration Server immediately available to an Nx_Port once it has successfully completed Fabric Login. However, the Fabric Configuration Server is not restricted or required to be part of a Fabric, and may be located in any Nx_Port.

6.2.2 Protocol

6.2.2.1 Overview

Fabric Configuration Server registration, deregistration and queries are managed through protocols containing a set of Request CT_IUs and Response CT_IUs supported by the Fabric Configuration Server.

For a Fabric Configuration Server request, the payload shall be transported from the requestor to the Fabric Configuration Server using a Request CT_IU. The corresponding Fabric Configuration Server response is transported from the Fabric Configuration Server to the requestor, in the Exchange established by the requestor, using a Response CT_IU.

The action of the Fabric Configuration Server is unaffected by Server Sessions.

6.2.2.2 CT_IU preamble values

The following values shall be set in the CT_IU preamble for Fabric Configuration Server request and their responses; fields not specified here shall be set as defined in 4.3.2:

a) GS_Subtype: as indicated in table 125; and

b) Command Code: see table 127 for Request command codes.

Table 127 – Fabric Configuration Server - Request Command Codes

Code	Mnemonic	Description	Reference subclause
0100h	GTIN	Get Topology Information	6.2.5.2
0101h	GIEL	Get Interconnect Element List	6.2.5.3
0111h	GIET	Get Interconnect Element Type	6.2.5.4
0112h	GDID	Get Domain Identifier	6.2.5.5
0113h	GMID	Get Management Identifier	6.2.5.6
0114h	GFN	Get Fabric Name	6.2.5.7

Table 127 – Fabric Configuration Server - Request Command Codes (Continued)

Code	Mnemonic	Description	Reference subclause
0115h	GIELN	Get Interconnect Element Logical Name	6.2.5.8
0116h	GMAL	Get Interconnect Element Management Address List	6.2.5.9
0117h	GIEIL	Get Interconnect Element Information List	6.2.5.10
0118h	GPL	Get Port List	6.2.5.11
0121h	GPT	Get Port Type	6.2.5.12
0122h	GPPN	Get Physical Port Number	6.2.5.13
0124h	GAPNL	Get Attached Port Name List	6.2.5.14
0126h	GPS	Get Port State	6.2.5.15
0127h	GPSC	Get Port Speed Capabilities	6.2.5.16
0128h	GATIN	Get Attached Topology Information	6.2.5.17
0129h	GPSCE	Get Port Speed Capabilities Extended	6.2.5.18
0130h	GSES	Get Switch Enforcement Status	6.2.5.19
0140h	GIEAG	Get Interconnect Element Attribute Group	6.2.5.20
0141h	GPAG	Get Port Attribute Group	6.2.5.21
0142h	GPAGE	Get Port Attribute Group Extended	6.2.5.22
0191h	GPLNL	Get Platform Node Name List	6.2.5.23
0192h	GPLT	Get Platform Type	6.2.5.24
0193h	GPLML	Get Platform Management Address List	6.2.5.25
0197h	GPAB	Get Platform Attribute Block	6.2.5.26
01A1h	GNPL	Get Platform Name - Node Name	6.2.5.27
01A2h	GPNL	Get Platform Name List	6.2.5.28
01A4h	GPFCP	Get Platform FCP Type	6.2.5.29
01A5h	GPLI	Get Platform OS LUN Mappings	6.2.5.30
01B1h	GNID	Get Node Identification Data - Node Name	6.2.5.31
0215h	RIELN	Register Interconnect Element Logical Name	6.2.5.32
0280h	RPL	Register Platform	6.2.5.33
0291h	RPLN	Register Platform Node Name	6.2.5.34

Table 127 – Fabric Configuration Server - Request Command Codes (Continued)

Code	Mnemonic	Description	Reference subclause
0292h	RPLT	Register Platform Type	6.2.5.35
0293h	RPLM	Register Platform Management Address	6.2.5.36
0298h	RPAB	Register Platform Attribute Block	6.2.5.37
029Ah	RPFCP	Register Platform FCP Type	6.2.5.38
029Bh	RPLI	Register Platform OS LUN Mappings	6.2.5.39
0380h	DPL	Deregister Platform	6.2.5.40
0391h	DPLN	Deregister Platform Node Name	6.2.5.41
0392h	DPLM	Deregister Platform Management Address	6.2.5.42
0393h	DPLML	Deregister Platform Management Address List	6.2.5.43
0394h	DPLI	Deregister Platform OS LUN Mappings	6.2.5.44
0395h	DPAB	Deregister Platform Attribute Block	6.2.5.45
039Fh	DPALL	De-Register All Platform Information	6.2.5.46
0400h	FTR	FC Trace Route	6.2.5.47
0401h	FPNG	FC Ping	6.2.5.48
other	reserved		

6.2.2.3 Registration

The registration requests defined for the Fabric Configuration Server are summarized in table 127. Some attributes do not have a corresponding registration request; this standard does not define the registration of those attributes.

The Fabric Configuration Server may reject registrations due to Fabric Configuration Server resource limitations. However, the Fabric Configuration Server shall support registration of all attributes, once registration of a single attribute has been accepted for a given Name_Identifier (see FC-FS-4).

The Fabric Configuration Server may reject any registration requests for reasons not specified in this document.

If overlapping registrations for the same attribute are performed, then the Fabric Configuration Server shall, when all registrations have completed, leave the attribute as one of the registered attribute values. However, it is indeterminate which of the overlapping registration requests take precedence.

6.2.2.4 Queries

The Fabric Configuration Server may reject any query requests for reasons not specified in this document. The queries defined for the Fabric Configuration Server are summarized in table 127.

6.2.3 Fabric Configuration Server Objects and Attributes

6.2.3.1 Overview

Figure 5 illustrates the physical Fabric, consisting of one or more Interconnect Elements, that each have some number of physical Ports (i.e., LCFs). These Ports are then connected either to other Ports on other Interconnect Elements, or to Nx_Ports outside of the physical Fabric.

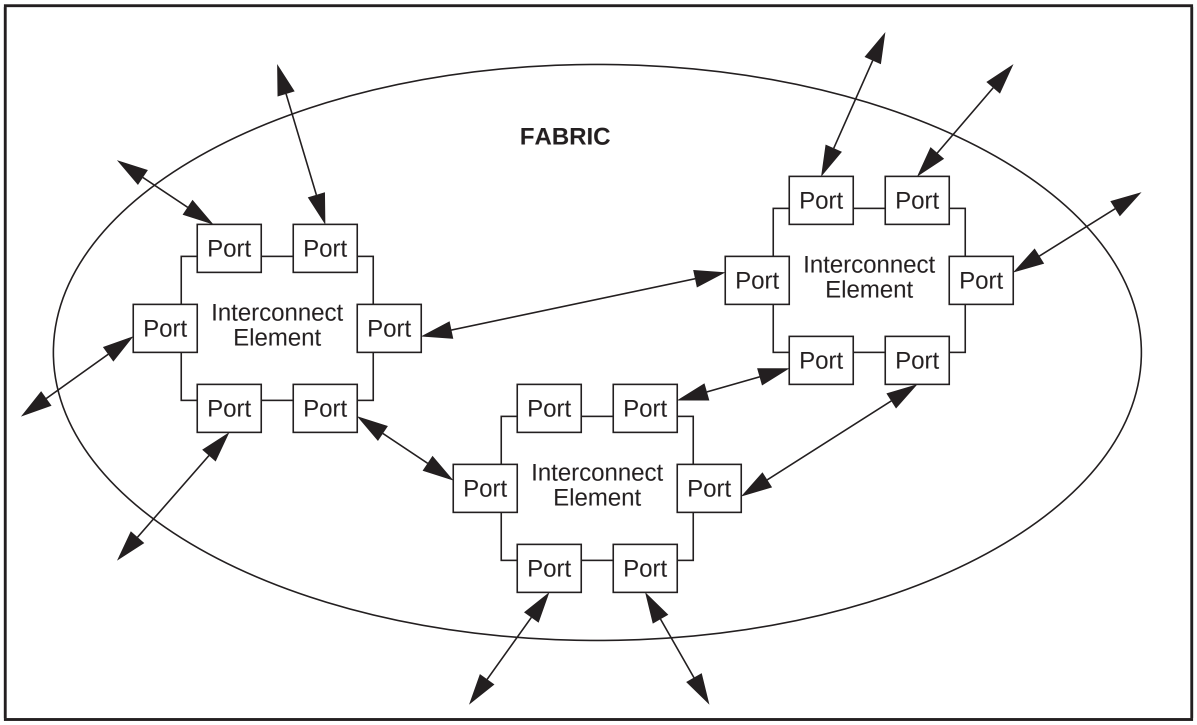


Figure 5 – Physical Fabric Illustration

The Fabric Configuration Server object model is shown in figure 6.

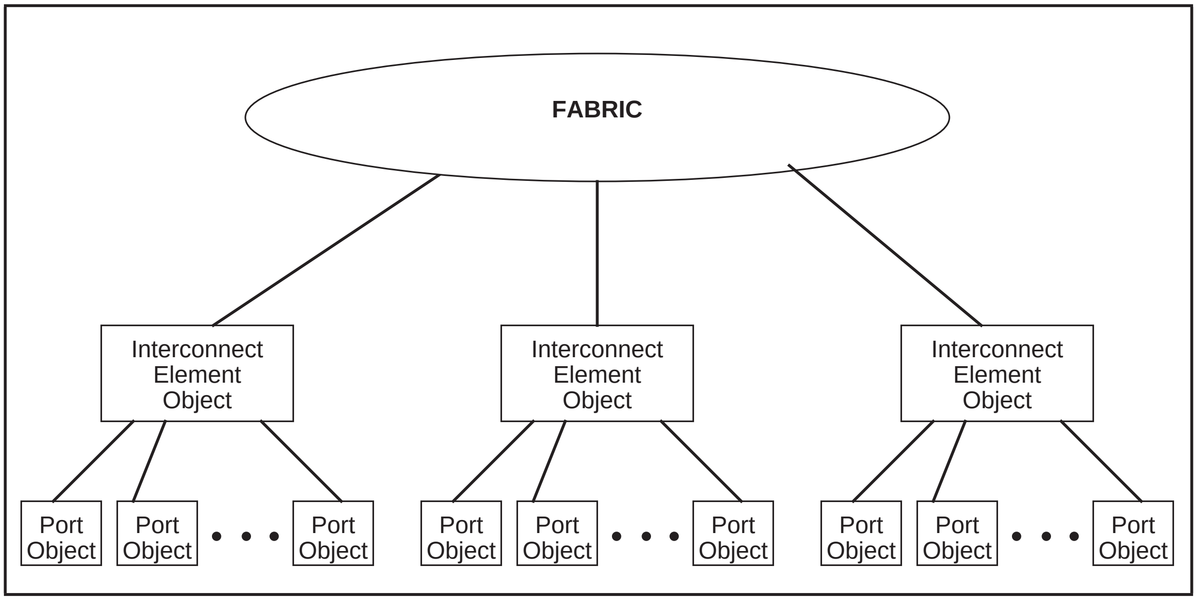


Figure 6 – Fabric Configuration Server Object Model

The base object class managed by the Fabric Configuration Server is the Interconnect Element object. Interconnect Element objects have one or more associated Port objects. One or more Interconnect Element objects belong to a Fabric. Interconnect Element objects and Port objects may have attributes associated with them, as shown in figure 7.

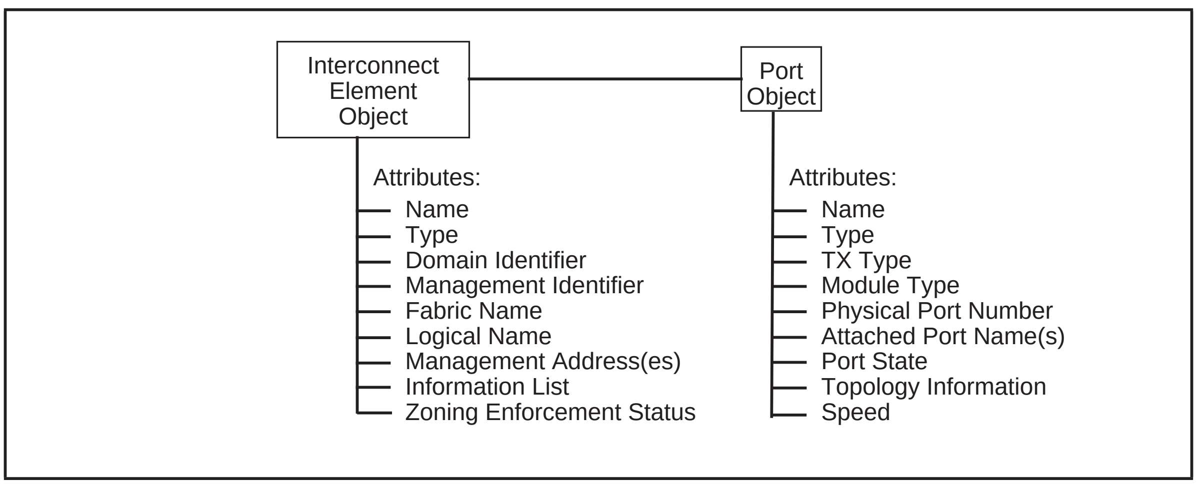


Figure 7 – Interconnect Element and Port attributes

6.2.3.2 Interconnect Element Object

6.2.3.2.1 Interconnect Element Name

The format of the Interconnect Element Name attribute shall be identical to the Name_Identifier format. If the Interconnect Element is a Switch (see FC-SW-6), the Interconnect Element Name attribute shall be the Switch_Name of the Switch.

This standard does not define how this attribute is registered with the Fabric Configuration Server. The null value for the Interconnect Element Name attribute is 00 00 00 00 00 00 00 00h.

6.2.3.2.2 Interconnect Element Type

The values of the Interconnect Element Type attribute shall be as shown in table 128.

Table 128 – Interconnect Element Type- encoding

Encoded value	Description
00h	Unknown
01h	Switch
02h	Hub
03h	Bridge
all others	Reserved

This standard does not define how this attribute is registered with the Fabric Configuration Server. The null Interconnect Element Type attribute value is set to ‘Unknown’.

6.2.3.2.3 Interconnect Element Domain Identifier

The format of the Interconnect Element Domain Identifier attribute shall be identical to the Domain Identifier format (see FC-SW-6).

This standard does not define how this attribute is registered with the Fabric Configuration Server. The null value for the Interconnect Element Domain Identifier attribute is 00h.

6.2.3.2.4 Interconnect Element Management Identifier

The format of the Interconnect Element Management Identifier attribute shall be identical to the address identifier format (see FC-FS-4). If the Interconnect Element is a Switch (see FC-SW-6), the Interconnect Element Management Identifier attribute shall be the Domain Controller identifier of the Switch.

This standard does not define how this attribute is registered with the Fabric Configuration Server. The null value for the Interconnect Element Management Identifier attribute is 00 00 00h.

6.2.3.2.5 Interconnect Element Fabric Name

The format of the Interconnect Element Fabric Name attribute shall be identical to the Name_Identifier format and the value shall be Fabric_Name (see FC-SW-6).

This standard does not define how this attribute is registered with the Fabric Configuration Server. The null value for the Interconnect Element Fabric Name attribute is 00 00 00 00 00 00 00 00h.

6.2.3.2.6 Interconnect Element Logical Name

The format of the Interconnect Element Logical Name attribute shall be as shown in table 129. The contents of these bytes are not defined and shall not be restricted by the Fabric Configuration Server.

Table 129 – Logical Name Format

Item	Size (Bytes)
Logical Name length (m)	1
Logical Name	m
Reserved	255-m

This attribute may be registered using the protocol described in 6.2.2.3. The null value for the Interconnect Element Logical Name attribute is a zero-length Interconnect Element Logical Name.

6.2.3.2.7 Interconnect Element Management Address

The format of the Interconnect Element Management Address attribute shall be as shown in table 130. Zero or more Management Address attributes shall be associated with an Interconnect Element object.

Table 130 – Management Address Format

Item	Size (Bytes)
Management Address length (m)	1
Management Address value	m
Reserved	255-m

The format of the Management Address shall use the format of the Uniform Resource Locator (URL) as defined in RFC2396, RFC1738 and RFC2732. The scheme field shall be as registered at http://www.iana.org/assignments/uri-schemes (see RFC2396). A null management address entry is specified as a Management Address length value of zero followed by 255 reserved bytes.

This standard does not define how this attribute is registered with the Fabric Configuration Server. The contents of the Management Address shall not be restricted by the Fabric Configuration Server.

NOTE 30 – Some legacy implementations may return a null management address of the format “snmp://0.0.0.0”. Applications should check for an IP address of 0.0.0.0 and if detected not use it.

6.2.3.2.8 Interconnect Element Information List

The format of the Interconnect Element Information List attribute shall be as shown in table 131. This standard does not define how this attribute is registered with the Fabric Configuration Server.

Table 131 – Information List Format

Item	Size (Bytes)
Reserved	3
List Length	1
Vendor name	w
Model name/number	x
Release code	y
Vendor-specific information	z

List Length (n): Specifies the length of the list in bytes, up to a maximum of 252.

Vendor name: A printable ASCII character string, terminated with a null (00h), that specifies the vendor name of the designated Interconnect Element.

Model name/number: A printable ASCII character string, terminated with a null (00h), that specifies the model name and/or model number of the designated Interconnect Element.

Release code: A printable ASCII character string, terminated with a null (00h), that specifies the release code or release level of the designated Interconnect Element.

Vendor-specific information: Zero or more printable ASCII character strings, each terminated with a null (00h), that contain other vendor-specific information regarding the designated Interconnect Element.

6.2.3.3 Port Object

6.2.3.3.1 Port Name

The format of the Port Name attribute shall be identical to the Name_Identifier format. The value of the Port Name attribute shall be the same as the value Port_Name in the Fabric Login ELS Accept payload.

This standard does not define how this attribute is registered with the Fabric Configuration Server. The null value for the Port Name attribute is 00 00 00 00 00 00 00 00h.

6.2.3.3.2 Port Type

The values of the Port Type attribute shall be as shown in table 132.

Table 132 – Port Type encoding

Encoded value	Description
00h	Unidentified
01h	N_Port
02h	NL_Port
03h	F/NL_Port
7Fh	Nx_Port
81h	F_Port
82h	FL_Port
84h	E_Port
85h	B_Port
86h	A_Port
C0h to FFh	Vendor Specific
all others	Reserved

This standard does not define how this attribute is registered with the Fabric Configuration Server. The null Port Type attribute value is set to ‘Unknown’.

6.2.3.3.3 Port TX Type

This attribute describes the technology that is incorporated in the transmitter of the module. The values of the Port TX Type attribute shall be as shown in table 133.

Table 133 – Port TX Type encoding

Encoded value	Description
01h	Unknown
02h	Long wave laser - LL (1 550 nm)
03h	Short wave laser - SN (850 nm)
04h	Long wave laser cost reduced - LC (1 310 nm)
05h	Electrical - EL
06h	10GBASE-SR 850nm laser[a]
07h	10GBASE-LR 1310nm laser[a]
08h	10GBASE-ER 1550nm laser[a]
09h	10GBASE-LX4 WWDM 1300nm laser[a]
0Ah	10GBASE-SW 850nm laser[a]
0Bh	10GBASE-LW 1310nm laser[a]
0Ch	10GBASE-EW 1550nm laser[a]
0Dh	10GBASE-CX4[a]
0Eh	Long wave laser - LZ (1 490 nm)
all others	Reserved

[a] See IEEE 802.3-2005.

This standard does not define how this attribute is registered with the Fabric Configuration Server. The null Port TX Type attribute value is set to ‘Unknown’.

6.2.3.3.4 Port Module Type

This attribute describes the form factor of the module. The values of the Port Module Type attribute shall be as shown in table 134.

Table 134 – Port Module Type encoding

Encoded value	Description
01h	Unknown
02h	Other
03h	Obsolete
04h	Embedded
05h	GLM
06h	GBIC with serial ID
07h	GBIC without serial ID
08h	SFP with Serial ID
09h	SFP without Serial ID
0Ah	XFP
0Bh	X2 Short
0Ch	X2 Medium
0Dh	X2 Tall
0Eh	XPAK Short

Table 134 – Port Module Type encoding(Continued)

Encoded value	Description
0Fh	XPAK Medium
10h	XPAK Tall
11h	XENPAK
12h	SFP-DWDM
13h	QSFP
14h	X2-DWDM
15h	CFP
16h	CFP2
17h	CFP4
18h	QSFP+
19h	QSFP28
all others	Reserved

This standard does not define how this attribute is registered with the Fabric Configuration Server. The null Port Module Type attribute value is set to 'Unknown'.

6.2.3.3.5 Physical Port Number

The format of the Physical Port Number attribute shall be as shown in table 135. The contents of this field are not defined and shall not be restricted by the Fabric Configuration Server, due to vendor specific methods for numbering physical ports.

Table 135 – Physical Port Number Format

Item	Size (Bytes)
Physical Port Number	4

This standard does not define how this attribute is registered with the Fabric Configuration Server. The null value for the Physical Port Number attribute is 00 00 00 00h.

6.2.3.3.6 Attached Port Name

The format of the Attached Port Name attribute shall be as shown in table 136. Zero or more Attached Port Name attributes may be associated with a Port object.

Table 136 – Attached Port Name Format

Item	Size (Bytes)
Port Name	8
Reserved	2
Port Flags	1
Port Type	1

Port Name: As defined in 6.2.3.3.1.

Port Flags: As shown in table 137.

Table 137 – Port Flags field bits

Bit Position	Description
7 to 2	Reserved
1	A value of one indicates that the Port supports the Get Topology Information Extended (GTIN) Link Service. A value of zero indicates that the Port does not support this ELS.
0	Obsolete

Port Type: As defined in 6.2.3.3.2.

This standard does not define how this attribute is registered with the Fabric Configuration Server. A Port object with a Port Type attribute value of “N_Port” or “NL_Port” shall have a null Attached Port Name List. The null value for the Attached Port Name List attribute shall be a zero length Attached Port Name List.

6.2.3.3.7 Port State

The values of the Port State attribute shall be as shown in table 138.

Table 138 – Port State encoding

Encoded value	Description
00h	Unknown
01h	Online - a frame may be passed through the FC_Port
02h	Offline - a frame is not able to be passed through the FC_Port
03h	Testing - FC_Port is in a test state
04h	Fault - FC_Port is not operational
E0h to FFh	Vendor specific
all others	Reserved

This standard does not define how this attribute is registered with the Fabric Configuration Server. The null Port State attribute value is set to ‘Unknown’.

6.2.3.3.8 Port Speed Capabilities object

The Port Speed Capabilities object identifies the data transfer rate capabilities of the LCF within the FC_Port. The format of the Port Speed Capabilities object shall be as shown in table 139.

Table 139 – Port Speed Capabilities object format

Item	Size (Bytes)
Port Speed Capabilities	2

All the LCF's potential data transfer speed operating points are indicated by setting the appropriate bit to one. More than one bit may be set at a time and are specified in table 140.

Table 140 – Port Speed Capabilities

Bit Position	Description	Bit Position	Description
15	1 GFC	7	40 GFC[a]
14	2 GFC	6	10 GE
13	4 GFC	5	40 GE
12	10 GFC[a]	4	100 GE
11	8 GFC	3	128 GFC
10	16 GFC	2	25 GE
9	20 GFC	1	Incomplete[b]
8	32 GFC	0	Unknown

[a] Legacy implementations may have used this bit for Ethernet.
[b] Complete Port Speed Capabilities information cannot be provided. The GPSCE command (see 6.2.5.18) may obtain additional Port Speed Capabilities.

6.2.3.3.9 Port Operating Speed object

The Port Operating Speed object identifies the current operating data transfer rate of the LCF within an FC_Port. The format of the Port Operating Speed object shall be as shown in table 141.

Table 141 – Port Operating Speed object format

Item	Size (Bytes)
Port Operating Speed	2

When a bit is set to one, it indicates the LCF is operating at the designated speed. Only one bit shall be set at a time. If the operating speed has not been established, then the “Speed not established” bit is set to one. If the LCF’s operating speed is not listed in table 142 or is not identifiable, then the “Unknown” bit is set to one. Valid bits are as shown in table 142.

Table 142 – Port Operating Speed

Bit Position	Description	Bit Position	Description
15	1 GFC	7	40 GFC[a]
14	2 GFC	6	10 GE
13	4 GFC	5	40 GE
12	10 GFC[a]	4	100 GE
11	8 GFC	3	128 GFC
10	16 GFC	2	25 GE
9	20 GFC	1	Unknown[b]
8	32 GFC	0	Speed not established

[a] Legacy implementations may have used this bit for Ethernet.
[b] The Port Operating Speed is not in this list. The GPSCE command (see 6.2.5.18) may obtain the Port Operating Speed.

6.2.3.3.10 Zoning Enforcement Status Object

The format of the Zoning Enforcement Status Object is depicted in table 143.

Table 143 – Zoning Enforcement Status Object

Item	Size (Bytes)
F_Port_Name	8
Port enforcement status	4

F_Port_Name: This field contains the F_Port_Name of the Fx_Port that the enforcement status object is referencing.

Port Enforcement Status: This is a 32 bit wide bit field that reports the actual enforcement status of the named Fx_Port. The defined bits are depicted in table 144.

Table 144 – Port Enforcement Status Bit Definitions

Bit	Interpretation
0	1 = Soft Zoning enforcement on 0 = Soft Zoning enforcement off
1	1 = Hard Zoning enforcement on 0 = Hard Zoning enforcement off
2	1 = Broadcast Zoning enforcement (see FC-SW-6) on 0 = Broadcast Zoning enforcement off
all others	Reserved

6.2.3.3.11 Port Speed Capabilities Extended object

The Port Speed Capabilities Extended object identifies the data transfer rate capabilities of the LCF within the FC_Port. The Port Speed Capabilities Extended object is formatted into multiple pages. The FC_Port returns the appropriate page that contains the applicable information for the FC_Port. Each page contains Speed Capabilities values and a Control field (see table 146) that the LCF may use to refine the scope of the capabilities returned. The Page bits in the Control field indicate which page the LCF is returning.

All the LCF's potential supported data transfer speeds are indicated by setting the appropriate bit in the Speed Capabilities field to one. More than one bit may be set at a time. The format of the Port Speed Capabilities Extended object is specified in table 145.

Table 145 – Port Speed Capabilities Extended object format

Item	Size (Bytes)
Speed Capabilities	3
Control	1

Speed Capabilities: supported data transfer speeds (see table 148).

Control: characteristics of the Port Speed Capabilities Extended object (see table 146).

Table 146 – Control field bits

Bit Position	Description
7 to 6	Reserved
5 to 4	Page
3	Parallel
2	Transport Type
1	Administratively configured
0	Unknown

Page: see table 147.

Table 147 – Page values

Encoded Values	Description
00b	Page zero
01b	Reserved
10b	Reserved
11b	Reserved

Parallel: a value of one indicates the link is operating in parallel lane mode. A value of zero indicates the link is operating in serial lane mode.

Transport Type: a value of one indicates the Ethernet transport type. A value of zero indicates the Fibre Channel transport type.

Administratively configured: a value of one indicates the capabilities have been administratively assigned. A value of zero indicates the capabilities have not been administratively assigned.

Unknown: a value of one indicates that the Speed Capabilities are not identifiable. A value of zero indicates that the Speed Capabilities are identifiable and provided in the Speed Capabilities field.

A list of references for Speed Capabilities object pages based on the Transport Type is shown in table 148.

Table 148 – Speed Capabilities based on Transport Type

Page	Transport Type	Reference
0	0 (i.e., Fibre Channel)	table 149
	1 (i.e., Ethernet)	table 150

The Port Speed Capabilities Extended page zero values for the Fibre Channel transport type are shown in table 149.

Table 149 – Port Speed Capabilities Extended Page Zero Fibre Channel values

Bit Position	Description	Bit Position	Description
31	1 GFC	15	Reserved Speed Capabilities
30	2 GFC	14	
29	4 GFC	13	
28	8 GFC	12	
27	10 GFC	11	
26	16 GFC	10	
25	20 GFC	9	
24	32 GFC	8	
23	40 GFC	7	Control (see table 146)
22	64 GFC	6	
21	128 GFC	5	
20	256 GFC	4	
19	Reserved Speed Capabilities	3	
18		2	
17		1	
16		0	

The Port Speed Capabilities Extended page zero values for the Ethernet transport type are shown in table 150.

Table 150 – Port Speed Capabilities Extended Page Zero Ethernet values

Bit Position	Description	Bit Position	Description
31	1 GE	15	Reserved Speed Capabilities
30	10 GE	14	
29	25 GE	13	
28	40 GE	12	
27	50 GE	11	
26	100 GE	10	
25	200 GE	9	
24	400 GE	8	
23	Reserved Speed Capabilities	7	Control (see table 146)
22		6	
21		5	
20		4	
19		3	
18		2	
17		1	
16		0	

6.2.3.3.12 Port Operating Speed Extended object

The Port Operating Speed Extended object identifies the current operating data transfer rate of the LCF within an FC_Port. The Port Operating Speed Extended object is formatted into multiple pages. The FC_Port returns the appropriate page that contains the applicable information for that FC_Port. Each page contains the Operating Speed value and a Control field (see table 152) that the LCF may use to refine the scope of the Operating Speed returned. The Page bits in the Control field indicate which page the LCF is returning.

If a bit in the Operating Speed field is set to one, it indicates the LCF is operating at the designated speed. Only one bit shall be set at a time. The format of the Port Operating Speeds Extended object is shown in table 151.

Table 151 – Port Operating Speed Extended object format

Item	Size (Bytes)
Operating Speed	3
Control	1

Operating Speed: possible operating data transfer speeds (see table 154).

Control: characteristics of the Port Operating Speed Extended object (see table 152).

Table 152 – Control field bits

Bit Position	Description
7 to 6	Reserved
5 to 4	Page
3	Parallel
2	Transport Type
1	Speed not established
0	Unknown

Page: see table 153.

Table 153 – Page values

Encoded Values	Description
00b	Page zero
01b	Reserved
10b	Reserved
11b	Reserved

Parallel: a value of one indicates the link is operating in parallel lane mode. A value of zero indicates the link is operating in serial lane mode.

Transport Type: a value of one indicates the Ethernet transport type. A value of zero indicates the Fibre Channel transport type.

Speed not established: a value of one indicates that the speed is not established (e.g., the LCF is not in the Active State (see FC-FS-6)). A value of zero indicates the speed is established.

Unknown: a value of one indicates the speed is not able to be provided (e.g., the LCF is in the Active State, but the Operating Speed cannot be determined). A value of zero indicates:

a) the speed is known and provided in the Operating Speed field; or

b) the speed is not established.

A list of references for Port Operating Speed object pages based on the Transport Type is shown in table 154.

Table 154 – Port Operating Speeds based on Transport Type

Page	Transport Type	Reference
0	0 (i.e., Fibre Channel)	table 155
	1 (i.e., Ethernet)	table 156

The Port Operating Speed Extended page zero values for the Fibre Channel transport type are shown in table 155.

Table 155 – Port Operating Speed Extended Page Zero Fibre Channel values

Bit Position	Description	Bit Position	Description
31	1 GFC	15	Reserved Operating Speed
30	2 GFC	14	
29	4 GFC	13	
28	8 GFC	12	
27	10 GFC	11	
26	16 GFC	10	
25	20 GFC	9	
24	32 GFC	8	
23	40 GFC	7	Control (see table 152)
22	64 GFC	6	
21	128 GFC	5	
20	256 GFC	4	
19	Reserved Operating Speed	3	
18		2	
17		1	
16		0	

The Port Operating Speed Extended page zero values for the Ethernet transport type are shown in table 156.

Table 156 – Port Operating Speed Extended Page Zero Ethernet values

Bit Position	Description	Bit Position	Description
31	1 GE	15	Reserved Operating Speed
30	10 GE	14	
29	25 GE	13	
28	40 GE	12	
27	50 GE	11	
26	100 GE	10	
25	200 GE	9	
24	400 GE	8	
23	Reserved Operating Speed	7	Control (see table 152)
22		6	
21		5	
20		4	
19		3	
18		2	
17		1	
16		0	

6.2.3.4 Platform Object

6.2.3.4.1 Overview

Platform objects are defined to provide the basic ability to associate one or more nodes with a single Platform for discovery and management. Platform objects may have attributes associated with them, as shown in figure 8.

Platform objects may support multiple FC-4 types, however, the only FC-4 specific Platform object currently defined is the FCP-4 Platform object as shown in figure 8.

The node attributes and the port attributes are identical to the name server objects (see 5.2).

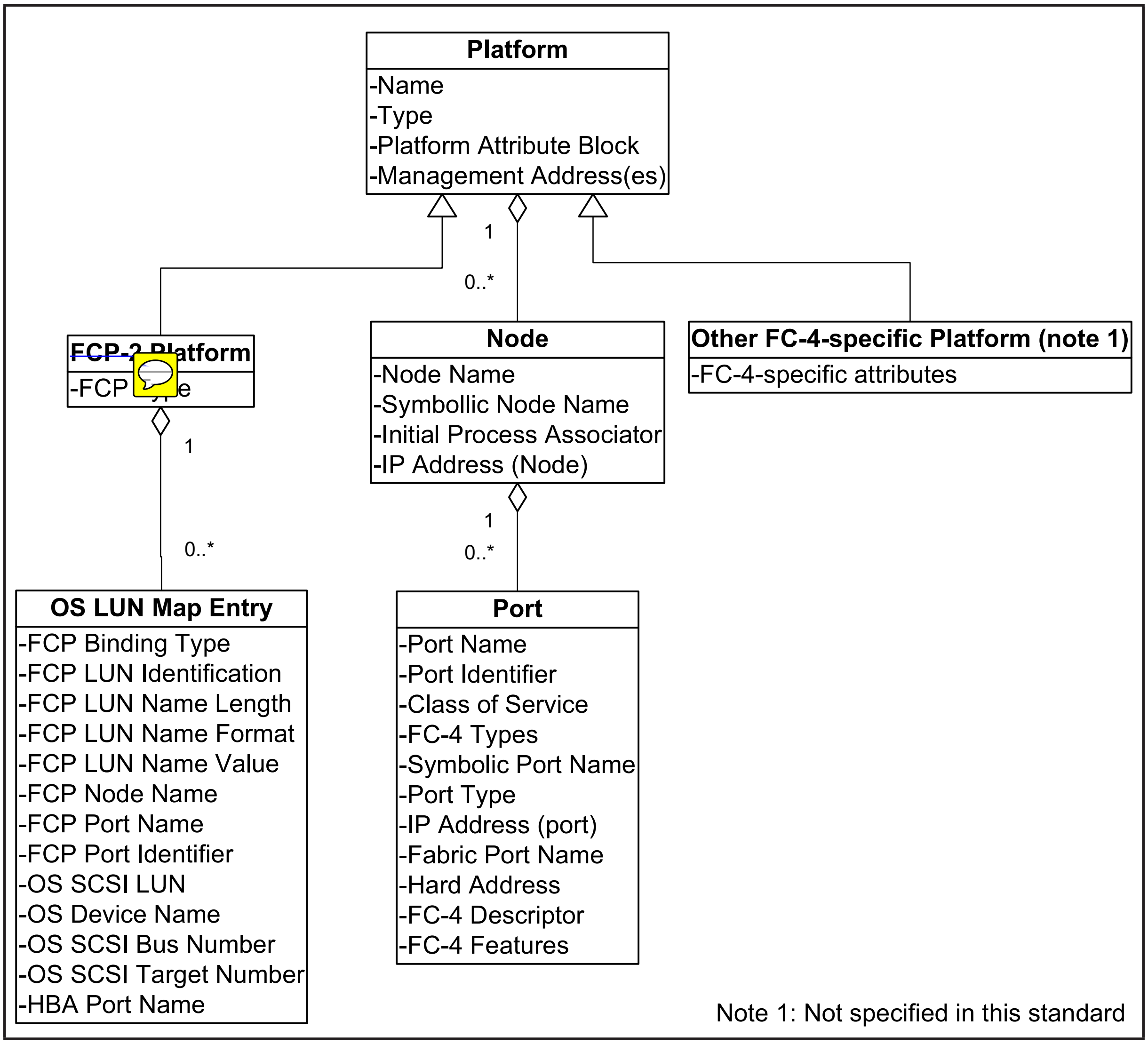


Figure 8 – Platform objects and attributes

6.2.3.4.2 Platform Name

The Platform Name attribute may be registered using the protocol described in 6.2.2.3. The null value for the Platform Name attribute is a zero-length Platform Name.

The format of the Platform Name attribute shall be as shown in table 157.

Table 157 – Platform Name Format

Item	Size (Bytes)
Platform Name length (m)	1
Platform Name	m
Platform Name Format	1
Reserved	254-m

The Platform Name length contains the length in bytes of the Platform Name. If the Platform Name length is equal to 255 then the Platform Name Format shall not be included in the Platform Name attribute and the Platform Name shall be 255 bytes and the Platform Name Format shall be assumed to be zero.

If the Platform Name Format equals zero then the Platform Name is not defined and shall not be restricted by the Fabric Configuration Server. If the Platform Name Format contains valid information the Platform Name format shall be as indicated by the Platform Name Format (see table 158).

The Platform Name Format contains a Platform Name Type field and a Code Set field that define the format of the Platform Name (see table 158).

Table 158 – Platform Name Format

Bit Byte	7	6	5	4	3	2	1	0
m+1	Platform Name Type				Code Set			

The Code Set field specifies the code set used for the Platform Name, as described in table 159.

Table 159 – Code set

Value	Description
0h	Reserved
1h	The Platform Name shall contain binary values.
2h	The Platform Name shall contain printable ASCII characters
3h to Fh	Reserved

The Platform Name Type field specifies the format and assignment authority for the Platform name, as described in table 160.

Table 160 – Platform Name type

Value	Description
0h	No assignment authority was used and consequently there is no guarantee that the Platform name is globally unique.
1h	The first eight bytes of the Platform Name field is a T10 Vendor ID (see 3.6). The organization associated with the T10 Vendor ID is responsible for ensuring that the remainder of the identifier field is unique.
2h	The Platform Name field contains an EUI-64. In this case, the identifier length field shall be set to eight. Note that the IEEE guidelines for EUI-64[a] specify a method for unambiguously encapsulating an EUI-48 within an EUI-64.
3h	The Platform Name field contains an FC-FS-4 Name_Identifier. Any FC-FS-4 identifier may be used, including one based on an IEEE Company_ID (CID).
4h	The Platform Name field contains a UUID. UUIDs are generated and formatted as described in RFC 4122.
5h to Fh	Reserved

[a] See https://standards.ieee.org/content/dam/ieee-standards/standards/web/documents/tutorials/eui.pdf.

6.2.3.4.3 Platform Type

The values of the Platform Type attribute shall be as shown in table 161.

Table 161 – Platform Type - encoding

Encoded value	Description
00 00 00 01h	Obsolete
00 00 00 02h	Other - none of the following
00 00 00 05h	Gateway
00 00 00 06h	Obsolete
00 00 00 07h	Obsolete
00 00 00 08h	Obsolete
00 00 00 09h	Obsolete
00 00 00 0Ah	Host
00 00 00 0Bh	Storage subsystem (see FC-LS-3)
00 00 00 0Ch	Obsolete
00 00 00 0Dh	Obsolete
00 00 00 0Eh	Storage access device (see FC-LS-3)
00 00 00 0Fh	Wavelength division multiplexer (see FC-LS-3)
00 00 00 11h	NAS server
00 00 00 12h	Bridge
00 00 00 13h	Virtualization device (see FC-LS-3)
00 00 00 14h	N_Port Virtualizer
xx xx xx FFh	Multi-function device (see table 162 for values to fill in for xx xx xx)
all others	Reserved

Table 162 – Multi-function device bit definitions

Bit position	Function
29	Gateway
27	Host
26	Storage subsystem
25	Storage access device
24	Wavelength division multiplexer
23	NAS server
22	Bridge
21	Virtualization device
20 to 8	Reserved

This attribute may be registered using the protocol described in 6.2.2.3. The null Platform Type attribute value is set to 00 00 02h ‘Other’.

6.2.3.4.4 Platform Attribute Block

The Platform Attribute Block is a variable length structure that contains attributes registered for the specified Platform. The format of the Platform Attribute Block is depicted in table 163.

Table 163 – Platform Attribute Block

Item	Size (Bytes)
Number of Platform Attribute Entries (n)	4
Platform Attribute Entry 1	w
Platform Attribute Entry 2	x
...	
Platform Attribute Entry n-1	y
Platform Attribute Entry n	z

Number of Platform Attribute Entries: This field specifies the number of Platform Attribute Entries contained in the Platform Attribute Block. This value shall be greater than or equal to one.

Platform Attribute Entry: A Platform Attribute Entry specifies a particular attribute registered with a Platform object.

6.2.3.4.5 Attribute Entry Format

The Fabric Configuration Server defines a general format to be used for attributes associated with Platform objects. The general format of the Attribute Entry is depicted in table 164.

Table 164 – Attribute Entry

Item	Size (Bytes)
Attribute Entry Type	2
Attribute Entry Length (n)	2
Attribute Entry Value	(see table 165)

Attribute Entry Type: This field indicates the Attribute Entry Type. Valid Attribute types are specific to the object to which they are associated. The Type codes are defined in table 165.

Attribute Entry Length: This field indicates the total length of the Attribute Entry. The total length in bytes shall be a multiple of four and includes the Attribute Entry Type, Attribute Entry Length, and Attribute Value fields.

Attribute Entry Value: This field specifies the Attribute Entry Value. Attribute Entry Values shall be at least four bytes in length and the length shall be a multiple of four. For variable length Attribute Value fields, fill bytes are added as necessary to the end of the actual value in order to ensure that the length of the value field is a multiple of four. Fill bytes shall be 00h. Attribute Entry types are defined in table 165.

Table 165 – Attribute Entry Types and their associated Values

Type	Value				
	Description	**Length (Bytes)**	**Type**	**Required**	**Multiples allowed[a]**
0001h	Vendor ID	12	ASCII	Yes	No
0002h	Product ID	20	ASCII	Yes	No
0003h	Product revision level	4 to 32	ASCII	No	No
0004h	Description	4 to128	ASCII	No	No
0005h	Label	4 to 64	ASCII	No	No
0006h	Location	4 to128	ASCII	No	No
0007h	System ID	4 to 64	ASCII	No	No
0008h	System management address	4 to128	ASCII	No	Yes
0009h	Cluster ID	4 to 64	ASCII	No	No
000Ah	Cluster management address	4 to 128	ASCII	No	Yes
000Bh	Supported FC-4 types	32	Binary	No	No
other values	Reserved				

[a] If a Platform Attribute Block contains multiple types for a type that does not allow multiples the command shall be rejected with a reason code of ‘Unable to perform command request’ and a reason code explanation of “Platform Attribute Block Contains Multiple Attributes of the Same Type”.

Vendor ID: An ASCII value that uniquely identifies the vendor. It is required that this value be the same as defined by the T10 SCSI Primary Commands -2 standard.

Product ID: An ASCII value that identifies the specific product and/or model for this vendor. It is required that this value be the same as defined by the T10 SCSI Primary Commands -2 standard.

Revision Level: An ASCII value that identifies the revision level for this Platform. It is required that this value be the same as defined by the T10 SCSI Primary Commands -2 standard.

Description: A textual description of the Platform.

Label: An administratively assigned symbolic name for the Platform.

Location: The physical location of the Platform (e.g., telephone closet, 3rd floor).

System ID: An identifier for a hosting system that this platform is associated with. This identifier is used to associate Platforms of logical types (e.g., logical partition) with a physical system. There is no requirement that this identifier be a fibre channel Name_Identifier.

System Management Address: A management address for the system. The format of this address is identical to the Interconnect Element Management Address attribute, and shall be as defined in 6.2.3.2.7.

Cluster ID: An identifier for a cluster that this Platform is associated with. Where a cluster is a set of independent Platforms that are managed together to provide increased performance capabilities, fail-over, etc. There is no requirement that this identifier be a fibre channel Name_Identifier.

Cluster Management Address: A management address for the cluster. The format of this address is identical to the Interconnect Element Management Address attribute, and shall be as defined in 6.2.3.2.7.

Supported FC-4 Types: This is an 8 word (256 bit) bit mask that indicates what FC-4 types are supported on this Platform (see 5.2.3.6). FCP-4 (FC-4 type 08h) is represented by bit 8 of word 0. The Fabric Configuration Server shall not attempt validation of the FC-4 Types attribute, and any value shall be accepted for this attribute.

6.2.3.4.6 Platform Node Name

The format of the Platform Node Name attribute shall be identical to the Name_Identifier format. Zero or more Platform Node Name attributes may be associated with a Platform object. Node_Names are registered to associate a Platform with the nodes.

This attribute may be registered using the protocol described in 6.2.2.3. The null value for the Platform Node Name attribute is 00 00 00 00 00 00 00 00h.

6.2.3.4.7 Platform Management Address

The format of the Platform Management Address attribute is identical to the Interconnect Element Management Address attribute, and shall be as defined in 6.2.3.2.7.

6.2.3.4.8 FCP-4 Platform Object

6.2.3.4.8.1 FCP-4 Platform Object Overview

The FCP-4 Platform may contain zero or more OS LUN Map Entries were each OS LUN Map Entry maps the operating systems representation of a LUN to the FCPs representation of a LUN.

6.2.3.4.8.2 FCP Type

This is a FC-4 specific attribute for Platforms that support FC-4 type of FCP-4. This a 4-byte encoded value, of which two bits are used to indicate the FCP-4 features for the Platform (see table 166).

Table 166 – FC-4 Specific Attributes

Bit Position	Description
31 to 4	Reserved
3 to 2	Reserved
1	1 = FCP initiator function supported (see INCITS T10 project FCP-4) 0 = Not supported
0	1 = FCP target function supported (see INCITS T10 project FCP-4) 0 = Not supported

6.2.3.4.9 OS LUN Map Entry Object

The OS LUN Map Entry contains a list of attributes that map the operating systems representation of a LUN to the FCPs representation of a LUN.

One or more OS LUN Map Entry attributes may be registered for Platforms that have the initiator bit set to a one in the FCP Type attribute. The OS LUN Map Entry contains Nx_Port information for both the mapped OS LUN and FCP LUN. It is based directly on the content of the HBA_FCPBindingEntry structure defined in the Common HBA API; however, it represents what is referenced as a Target Mapping in the Common HBA API (see FC-HBA). If there is an OS LUN Map Entry then all the attributes are required, however, some of the attributes may contain null values (i.e., hex zeros) (see table 167).

Table 167 – OS LUN Map Entry format

Item	Size (Bytes)	Required	Null Value (if not required)
FCP Binding Type	4	Yes	
FCP LUN	8	Yes	
FCP LUID Length	2	Yes	
Reserved	2	Yes	
FCP LUID	8 to 256	Yes	
FCP Node Name	8	Yes	
FCP Port Name	8	Yes	
Reserved	1	Yes	
FCP Port Identifier	3	Yes	
OS Device Name Length (y)[a]	4	Yes	
OS Device Name	y	Yes	
OS SCSI Bus Number	4	No	0000 0000h
OS SCSI Target Number	4	No	0000 0000h
OS SCSI LUN	4	Yes	
HBA Port Name	8	Yes	

[a] The OS Device Name Length (y) field includes the length of the OS Device Name, plus 4 bytes for the field proceeding the OS Device Name field (i.e., OS Device Name Length (y)).

FCP Binding Type: The FCP Binding Type attribute describes the type of binding for the LUN mapping, or if the FCP LUN is unmapped. All values except zero (i.e., not mapped) are defined by the Common HBA API HBA_Bind_Type declaration (see FC-HBA).

FCP LUN: The 64-bit SCSI LUN of a SCSI logical unit accessed by a SCSI Service Delivery Subsystem (see SAM-3).

FCP LUID Length: The length in bytes of the FCP LUID field plus four.

FCP LUID: An identification descriptor of association zero for the logical unit to which the OS LUN maps (see ~~SPC-3~~). If the length of the identification descriptor is not a multiple of four bytes, it shall be padded with trailing zero bytes to the next multiple of four bytes. Any necessary padding shall be reflected in the value of the FCP LUID Length field but shall not cause adjustment to the length field embedded in the FCP LUID. A Platform that supports FC-HBA Target Mapping (see FC-HBA) shall register the same identification descriptor for a mapping as it returns via the FC-HBA interface, varying only by padding required by this subclause.

FCP Node Name: The Name_Identifier (see FC-FS-4) of the node in the target device that the initiator OS LUN maps to.

FCP Port Name: The Name_Identifier (see FC-FS-4) of the Nx_Port in the target device that the initiator OS LUN maps to.

FCP Port Identifier: The port identifier of the Nx_Port in the target device that the initiator OS LUN maps to.

OS Device Name Length (y): The length (in bytes) of the OS Device Name. The OS Device Name Length (y) field includes the length of the OS Device Name, plus 4 bytes for the field proceeding the OS Device Name field (i.e., OS Device Name Length (y))

OS Device Name: This is a symbolic device name assigned for the target device by the HBA device driver.

Example: /dev/sd3

OS SCSI Bus Number: A SCSI bus number assigned for the target device by the HBA device driver.

OS SCSI Target Number: A SCSI target number assigned for the target device by the HBA device driver.

OS SCSI LUN: A SCSI LUN assigned to the logical unit on the target device by the HBA device driver.

HBA Port Name: The HBA N_Port_Name (see FC-FS-4) that corresponds to this LUN mapping.

6.2.3.4.10 Platform Description

A textual description of the Platform. This value should include the full name and version identification of the Platform’s hardware type and software operating system. The Platform Description shall only contain printable ASCII characters.

6.2.3.4.11 Platform Label

An administratively assigned symbolic name for the Platform. The Platform Label shall only contain printable ASCII characters.

6.2.3.4.12 Platform Location

The physical location of the Platform (e.g., telephone closet, 3rd floor). The Platform Location shall only contain printable ASCII characters.

6.2.4 Reason code Explanations

A Reject CT_IU (see 4.4.4) shall notify the requestor that the request has been unsuccessfully completed. The first error condition encountered shall be the error reported by the Reject CT_IU.

If a Fabric Configuration Server request is not supported, the request shall be rejected with a Reject CT_IU reason code of “Command not supported” and a reason code explanation of “No additional explanation”.

If a valid Fabric Configuration Server request is not received, the request is rejected with a reason code of “Invalid command code” and a reason code explanation of “No additional explanation”.

If a Fabric Configuration Server request is rejected with a reason code of ‘Unable to perform command request’, then one of the reason code explanations, shown in table 168, are returned.

Table 168 – Reject CT_IU Reason code Explanations

Encoded value	Description
00h	No additional explanation
01h	Invalid Name_Identifier for Interconnect Element or Port
10h	Interconnect Element List not available
11h	Interconnect Element Type not available
12h	Domain Identifier not available
13h	Management Identifier not available
14h	Fabric Name not available
15h	Interconnect Element Logical Name not available
16h	Management Address List not available
17h	Interconnect Element Information List not available
30h	Port List not available
31h	Port Type not available
32h	Physical Port Number not available
34h	Attached Port Name List not available
36h	Port State not available
50h	Unable To register Interconnect Element Logical Name
60h	Platform Name does not exist
61h	Platform Name already exists.
62h	Platform Node Name does not exist
63h	Platform Node Name already exists.
64h to 6Fh	Vendor Specific
70h	Platform register operation failed - resource unavailable.

Table 168 – Reject CT_IU Reason code Explanations (Continued)

Encoded value	Description
71h	Zero entries in OS LUN Map
72h	Invalid OS Device Name Length
73h	Platform Attribute Block Contains Multiple Attributes of the Same Type
74h	Invalid Platform Attribute Block Length
75h	Required Platform Attributes Not Present
76h	Command already in progress
Others	Reserved

If a Fabric Configuration Server Query request other than GIEL and GPL is rejected by the Fabric Configuration Server because the attribute specified in the request is not found in the Fabric Configuration Server data base, then the Reject CT_IU reason code shall be ‘Unable to perform command request’, with a reason code explanation that indicates the specified attribute is not available.

6.2.5 Commands

6.2.5.1 Overview

The commands defined for the Fabric Configuration Server are summarized in table 127.

6.2.5.2 Query - Get Topology Information (GTIN)

The Fabric Configuration Server shall, if it receives a GTIN request, return topology information for the specified Interconnect Element Domain. The format of the GTIN Request CT_IU is shown in table 169.

Table 169 – GTIN Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Reserved	5
Interconnect Element Domain ID	1
Reserved	6

If the GTIN request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the Accept CT_IU to a GTIN request is shown in table 170, table 171, and table 172.

Table 170 – Accept CT_IU to GTIN Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Reserved	8
Interconnect Element Name	8
Reserved	8
Interconnect Element Domain ID	1
Reserved	3
Reserved	1
Number of Interconnect Elements	1
Number of Ports on the Interconnection Element (i.e., E_Ports, F_Ports, and FL_Ports)	2
Interconnect Element Domain ID #1	1
...	
Interconnect Element Domain ID #n (n = Number of Interconnect Elements)	1
Pad (Padded to word boundary)	0,1,2, or 3
Topology Information Descriptor #1 (see table 171)	68 + (72 x Number of attached Ports)
...	
Topology Information Descriptor #n (n = Number of Fx_Ports)	68 + (72 x Number of attached Ports)

Table 171 – Topology Information Descriptor

Item	Size (Bytes)
Interconnect Element Port Name	8
Vendor Specific	8
Vendor Specific	16
Reserved	3
Interconnect Element Port Type (see table 132)	1
Interconnect Element Physical Port Number	4
Number of Attached Ports	4
Vendor Specific	24
Attached Port Descriptor #1	72
...	
Attached Port Descriptor #m (m = Number of Attached Ports)	72

Table 172 – Attached Port Descriptor

Item	Size (Bytes)
Vendor Specific	4
Attached Port Name	8
Attached Node Name	8
Vendor Specific	16
Reserved	3
Attached Port Type (see table 132)	1
Attached Physical Port Number (see 6.2.3.3.5))	4
Vendor Specific	28

6.2.5.3 Query - Get Interconnect Element List (GIEL)

The Fabric Configuration Server shall, if it receives a GIEL request, return all Interconnect Element Names in the Fabric. The format of the GIEL Request CT_IU is shown in table 173.

Table 173 – GIEL Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

If the GIEL request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the Accept CT_IU to a GIEL request is shown in table 174.

Table 174 – Accept CT_IU to GIEL Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Number of Interconnect Element entries (n)	4
Interconnect Element Name #1	8
Reserved	3
Interconnect Element Type #1	1
Interconnect Element Name #2	8
Reserved	3
Interconnect Element Type #2	1
...	
Interconnect Element Name #n	8
Reserved	3
Interconnect Element Type #n	1

One or more Interconnect Element entries are returned, and the Interconnect Element entries may be returned in any order, and the order may be different for every request even if the same Interconnect Element entries are returned and the requestor is the same.

6.2.5.4 Query - Get Interconnect Element Type (GIET)

The Fabric Configuration Server shall, if it receives a GIET request, return the Interconnect Element Type for the specified Interconnect Element Name. The format of the GIET Request CT_IU is shown in table 175.

Table 175 – GIET Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Interconnect Element Name	8

If the GIET request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the Accept CT_IU to a GIET request is shown in table 176.

Table 176 – Accept CT_IU to GIET Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Reserved	3
Interconnect Element Type	1

6.2.5.5 Query - Get Interconnect Element Domain Identifier (GDID)

The Fabric Configuration Server shall, if it receives a GDID request, return the Interconnect Element Domain Identifier for the specified Interconnect Element Name. The format of the GDID Request CT_IU is shown in table 177.

Table 177 – GDID Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Interconnect Element Name	8

If the GDID request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the Accept CT_IU to a GDID request is shown in table 178.

Table 178 – Accept CT_IU to GDID Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Reserved	1
Interconnect Element Domain Identifier	1
Reserved	2

6.2.5.6 Query - Get Interconnect Element Management Identifier (GMID)

The Fabric Configuration Server shall, if it receives a GMID request, return the Interconnect Element Management Identifier for the specified Interconnect Element Name. The format of the GMID Request CT_IU is shown in table 179.

Table 179 – GMID Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Interconnect Element Name	8

If the GMID request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the Accept CT_IU to a GMID request is shown in table 180.

Table 180 – Accept CT_IU to GMID Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Reserved	1
Interconnect Element Management Identifier	3

6.2.5.7 Query - Get Interconnect Element Fabric Name (GFN)

The Fabric Configuration Server shall, if it receives a GFN request, return the Interconnect Element Fabric Name for the specified Interconnect Element Name. The format of the GFN Request CT_IU is shown in table 181.

Table 181 – GFN Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Interconnect Element Name	8

If the GFN request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the Accept CT_IU to a GFN request is shown in table 182.

Table 182 – Accept CT_IU to GFN Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Interconnect Element Fabric Name	8

6.2.5.8 Query - Get Interconnect Element Logical Name (GIELN)

The Fabric Configuration Server shall, if it receives a GIELN request, return the Interconnect Element Logical Name for the specified Interconnect Element Name. The format of the GIELN Request CT_IU is shown in table 183.

Table 183 – GIELN Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Interconnect Element Name	8

If the GIELN request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the Accept CT_IU to a GIELN request is shown in table 184.

Table 184 – Accept CT_IU to GIELN Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Interconnect Element Logical Name	256

6.2.5.9 Query - Get Interconnect Element Management Address List (GMAL)

The Fabric Configuration Server shall, if it receives a GMAL request, return all Interconnect Element Management Address attributes for the specified Interconnect Element Name. The format of the GMAL Request CT_IU is shown in table 185.

Table 185 – GMAL Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Interconnect Element Name	8

If the GMAL request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the Accept CT_IU to a GMAL request is shown in table 186.

Table 186 – Accept CT_IU to GMAL Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Number of Management Address entries (n)	4
Management Address #1	256
Management Address #2	256
...	
Management Address #n	256

One or more Interconnect Element Management Address entries are returned, and the entries may be returned in any order, and the order may be different for every request even if the same entries are returned and the requestor is the same.

If no management addresses are supported, the Number of Management Address entries field shall be set to zero.

If management addresses are supported but no management address entries are registered, the Number of Management Address entries field shall be set to one, and one null management address entry (see 6.2.3.2.7) shall be returned.

6.2.5.10 Query - Get Interconnect Element Information List (GIEIL)

The Fabric Configuration Server shall, if it receives a GIEIL request, return the Interconnect Element Information List for the specified Interconnect Element Name. The format of the GIEIL Request CT_IU is shown in table 187.

Table 187 – GIEIL Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Interconnect Element Name	8

If the GIEIL request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the Accept CT_IU to a GIEIL request is shown in table 188.

Table 188 – Accept CT_IU to GIEIL Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Interconnect Element Information List	m
Reserved	256-m

6.2.5.11 Query - Get Port List (GPL)

The Fabric Configuration Server shall, if it receives a GPL request, return all Port Names and their associated Port Types, Port TX Types, and Port Module Types, for the specified Interconnect Element Name. The format of the GPL Request CT_IU is shown in table 189.

Table 189 – GPL Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Interconnect Element Name	8

If the GPL request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the Accept CT_IU to a GPL request is shown in table 190.

Table 190 – Accept CT_IU to GPL Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Number of Port entries (n)	4
Port Name #1	8
Reserved	1
Port Module Type #1	1
Port TX Type #1	1
Port Type #1	1
Port Name #2	8
Reserved	1
Port Module Type #2	1
Port TX Type #2	1
Port Type #2	1
...	
Port Name #n	8
Reserved	1
Port Module Type #n	1
Port TX Type #n	1
Port Type #n	1

One or more Port entries are returned, and the entries may be returned in any order, and the order may be different for every request even if the same entries are returned and the requestor is the same.

6.2.5.12 Query - Get Port Type (GPT)

The Fabric Configuration Server shall, if it receives a GPT request, return the Port Type for the specified Port Name. The format of the GPT Request CT_IU is shown in table 191.

Table 191 – GPT Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Port Name	8

If the GPT request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the Accept CT_IU to a GPT request is shown in table 192.

Table 192 – Accept CT_IU to GPT Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Reserved	3
Port Type	1

6.2.5.13 Query - Get Physical Port Number (GPPN)

The Fabric Configuration Server shall, if it receives a GPPN request, return the Physical Port Number for the specified Port Name. The format of the GPPN Request CT_IU is shown in table 193.

Table 193 – GPPN Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Port Name	8

If the GPPN request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the Accept CT_IU to a GPPN request is shown in table 194.

Table 194 – Accept CT_IU to GPPN Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Physical Port Number	4

6.2.5.14 Query - Get Attached Port Name List (GAPNL)

The Fabric Configuration Server shall, if it receives a GAPNL request, return all Attached Port Name attributes for the specified Port Name. The format of the GAPNL Request CT_IU is shown in table 195.

Table 195 – GAPNL Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Port Name	8

If the GAPNL request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the Accept CT_IU to a GAPNL request is shown in table 196.

Table 196 – Accept CT_IU to GAPNL Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Number of Attached Port entries (n)	4
Attached Port Name #1	12
Attached Port Name #2	12
...	
Attached Port Name #n	12

One or more Attached Port entries are returned, and the entries may be returned in any order, and the order may be different for every request even if the same entries are returned and the requestor is the same.

6.2.5.15 Query - Get Port State (GPS)

The Fabric Configuration Server shall, if it receives a GPS request, return the Port Type and Port State for the specified Port Name. The format of the GPS Request CT_IU is shown in table 197.

Table 197 – GPS Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Port Name	8

If the GPS request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the Accept CT_IU to a GPS request is shown in table 198.

Table 198 – Accept CT_IU to GPS Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Reserved	3
Port Type	1
Reserved	3
Port State	1

6.2.5.16 Query - Get Port Speed Capabilities (GPSC)

When the Fabric Configuration Server receives a Get Port Speed Capabilities (GPSC) request, it shall get the current Port Operating Speed and Port Speed Capabilities. The format of the GPSC request is shown in table 199.

Table 199 – GPSC Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Port Name	8

If the GPSC request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the Accept CT_IU to a GPSC is shown in table 200.

Table 200 – GPSC Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Port Speed Capabilities	2
Port Operating Speed	2

6.2.5.17 Query - Get Attached Topology Information (GATIN)

The Fabric Configuration Server shall, if it receives a GATIN request, return the topology information descriptor (see table 171) for the specified Port Name. The format of the GATIN Request CT_IU is shown in table 201.

Table 201 – GATIN Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Interconnect Element Port Name	8

If the GATIN request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the Accept CT_IU to a GATIN request is shown in table 202.

Table 202 – Accept CT_IU to GATIN Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Topology Information Descriptor (see table 171)	68 + (72 x Number of attached Ports)

NOTE 31 – This request may be used to discover topologies “behind” a Port, such as arbitrated loops or bridges to other interconnects. A typical approach might be to collect all of the Ports within an Interconnect Element using GPL (see 6.2.5.11), then issue this command to each Port in turn to discover the additional topologies.

6.2.5.18 Query - Get Port Speed Capabilities Extended (GPSCE)

When the Fabric Configuration Server receives a Get Port Speed Capabilities Extended (GPSCE) request, it shall get the current Port Operating Speed and Port Speed Capabilities. The format of the GPSCE request is shown in table 203.

Table 203 – GPSCE Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Port Name	8

If the GPSCE request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the Accept CT_IU to a GPSCE is shown in table 204.

Table 204 – GPSCE Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Port Speed Capabilities Extended object	4
Port Operating Speed Extended object	4

6.2.5.19 Query - Get Switch Enforcement Status (GSES)

The Zoning enforcement status of an Fx_Port is implicitly managed through the Fabric Zone Server by using the Hard Zoning or the Broadcast Zoning attributes, but it may be explicitly read through the Fabric Configuration Server by using the Get Switch Enforcement Status (GSES) request.

The Fabric Configuration Server shall, if it receives a GSES request, return the actual Zoning enforcement status for each Fx_Port of the specified Switch. The GSES request payload shall specify the Switch Name identifying the Switch for which the Zoning enforcement status information is sought. The GSES accept payload contains the Zoning enforcement status for each Fx_Port of the Switch specified in the request.

If the GSES request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the GSES request payload is depicted in table 205.

Table 205 – GSES Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Interconnect Element Name	8

Interconnect Element Name: The Name of the Interconnect Element that the enforcement status is being sought.

The format of the GSES accept payload is depicted in table 206.

Table 206 – GSES Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Number of Zoning enforcement status objects (n)	4
Zoning enforcement status object #1	see table 143
Zoning enforcement status object #2	see table 143
...	
Zoning enforcement status object #n	see table 143

6.2.5.20 Query -- Get Interconnect Element Attribute Group (GIEAG)

The Fabric Configuration Server shall, if it receives a GIEAG request, return all the Interconnect Element attributes defined in this standard. The format of the GIEAG request is shown in table 207.

Table 207 – GIEAG Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Interconnect Element Name	8

If the GIEAG request is not supported a Reject CT_IU shall be returned and the reason code should be Command Not Supported, no further explanation.

If a GIEAG request contains an Interconnect Element name that is no longer associated with the Fabric, a Reject CT_IU with the reason code Unable to Perform Command, and the reason code explanation of Invalid Name_Identifier for the Interconnect Element or Port shall be returned (see table 168).

If no attributes are available for the interconnect element name then all the defined GIEAG Accept Invalid Field bits set shall be returned and the attribute fields shall be filled with the null value appropriate for each Interconnect Element attribute. If any particular Interconnect Element attribute can not be returned, the field Invalid Field Flags bit corresponding to that attribute shall be set. In this way, it is possible to get partial information about an Interconnect Element if other information is not available.

The format of the Accept CT_IU to a GIEAG request is shown in table 208.

Table 208 – GIEAG Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Invalid Field Flags	2
Interconnect Element Type	1
Interconnect Element Domain Identifier	1
Reserved	1
Interconnect Element Management Identifier	3
Interconnect Element Fabric Name	8
Interconnect Element Logical Name	256
Interconnect Element Management Address	256
Interconnect Element Information List	m
Reserved	256-m

Invalid Field Flags: The Invalid Field Flags shall be defined as shown in table 209.

Table 209 – Invalid Field Flags

Bit	Description	Value
0	Invalid Interconnect Element Type	1 = The Interconnect Element Type field is invalid and the field is set to the NULL value for that field. 0 = Field is valid.
1	Invalid Interconnect Element Domain Identifier	1 = The Interconnect Element Domain Identifier field is invalid and the field is set to the NULL value for that field. 0 = Field is valid.
2	Invalid Interconnect Element Management Identifier	1 = The Interconnect Element Management Identifier field is invalid and the field is set to the NULL value for that field. 0 = Field is valid.
3	Invalid Interconnect Element Fabric Name	1 = The Interconnect Element Fabric Name field is invalid and the field is set to the NULL value for that field. 0 = Field is valid.
4	Invalid Logical Name	1 = The Interconnect Element Logical Name field is invalid and the field is set to the NULL value for that field. 0 = Field is valid.
5	Invalid Management Address	1 = The Interconnect Element Management Address field is invalid and the field is set to the NULL value for that field. 0 = Field is valid.
6	Invalid Information List	1 = The Interconnect Element Information List field is invalid and the field is set to the NULL value for that field. 0 = Field is valid.
7 to 15	Reserved	

6.2.5.21 Query -- Get Port Attribute Group (GPAG)

The Fabric Configuration Server shall, if it receives a GPAG request, return a group of Port attributes defined in this standard for the designated Port Names. The format of the GPAG request is shown in table 210.

Table 210 – GPAG Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Interconnect Element Name	8
Number of Port Names	4
Port Name 1	8
...	
Port Name n	8

If the GPAG request is not supported a Reject CT_IU shall be returned and the reject reason should be Command Not Supported, no further explanation.

If a GPAG request contains an Interconnect Element name that is no longer associated with the Fabric, a Reject CT_IU with the reason code Unable to Perform Command, and the reason code explanation of Invalid Name_Identifier for the Interconnect Element or Port shall be returned (see table 168).

If a GPAG request contains a Port Name that is no longer associated with the Interconnect Element Name specified in the request, the Port Name with all the Port Attribute Group field invalid bits set shall be returned and the “Port Name not found” bit in the Invalid Field Flags will be set and the attribute group fields shall be filled with the null value appropriate for each attribute. If any particular Port attribute can not be returned, the field invalid bit corresponding to that attribute shall be set. In this way, it is possible to get partial information about an FC_Port if other information is not available.

The format of the Port Attribute Group is shown in table 211.

Table 211 – Port Attribute Group

Item	Size (Bytes)
Reserved	1
Invalid Field Flags	2
Port Type	1
Reserved	3
Port State	1
Physical Port Number	4
Port Speed Capabilities~~[a]~~	2
Port Operating Speed~~[b]~~	2

Invalid Field Flags: The Invalid Field Flags field shall be set as shown in table 212.

Table 212 – Invalid Field Flags

Bit	Description	Value
0	Port Name Not Found	1 = The Port Name was not associated with the Interconnect Element Name. 0 = At least one valid attribute is returned for the Port Name.
1	Invalid Port Type	1 = The Port Type field is invalid and the field is set to the NULL value for that field. 0 = Field is valid.
2	Invalid Port State	1 = The Port State field is invalid and the field is set to the NULL value for that field. 0 = Field is valid.
3	Invalid Physical Port Number	1 = The Physical Port Number field is invalid and the field is set to the NULL value for that field. 0 = Field is valid.
4	Invalid Port Speed Capabilities	1 = The Port Speed Capabilities field is invalid and the field is set to the NULL value for that field. 0 = Field is valid.
5	Invalid Port Operating Speed	1 = The Port Operating Speed field is invalid and the field is set to the NULL value for that field. 0 = Field is valid.
6 to 15	Reserved	

The format of the GPAG Accept CT_IU is shown in table 213.

Table 213 – GPAG Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Number of Port Names	4
Port Name 1	8
Port Attribute Group 1	16
...	
Port Name n	8
Port Attribute Group n	16

6.2.5.22 Query -- Get Port Attribute Group Extended (GPAGE)

The Fabric Configuration Server shall, if it receives a GPAGE request, return a group of Port attributes defined in this standard for the designated Port Names. The format of the GPAGE request is shown in table 214.

Table 214 – GPAGE Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Interconnect Element Name	8
Number of Port Names	4
Port Name 1	8
...	
Port Name n	8

If the GPAGE request is not supported a Reject CT_IU shall be returned and the reject reason should be Command Not Supported, no further explanation.

If a GPAGE request contains an Interconnect Element name that is no longer associated with the Fabric, a Reject CT_IU with the reason code Unable to Perform Command, and the reason code explanation of Invalid Name_Identifier for the Interconnect Element or Port shall be returned (see table 168).

If a GPAGE request contains a Port Name that is no longer associated with the Interconnect Element Name specified in the request, the Port Name with all the Port Attribute Group field invalid bits set shall be returned and the “Port Name not found” bit in the Invalid Field Flags will be set and the attribute group fields shall be filled with the null value appropriate for each attribute. If any particular

Port attribute can not be returned, the field invalid bit corresponding to that attribute shall be set. In this way, it is possible to get partial information about an FC_Port if other information is not available.

The format of the Port Attribute Group is shown in table 215.

Table 215 – Port Attribute Group

Item	Size (Bytes)
Reserved	1
Invalid Field Flags	2
Port Type	1
Reserved	3
Port State	1
Physical Port Number	4
Port Speed Capabilities Extended object	4
Port Operating Speed Extended object	4

Invalid Field Flags: The Invalid Field Flags field shall be set as shown in table 212.

The format of the GPAGE Accept CT_IU is shown in table 216.

Table 216 – GPAGE Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Number of Port Names	4
Port Name 1	8
Port Attribute Group 1	~~16~~
...	
Port Name n	8
Port Attribute Group n	~~16~~

6.2.5.23 Query - Get Platform Node Name List (GPLNL)

The Fabric Configuration Server shall, if it receives a GPLNL request, return all Node Name attributes for the specified Platform Name. The format of the GPLNL Request CT_IU is shown in table 217.

Table 217 – GPLNL Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Platform Name	256

If the GPLNL request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the Accept CT_IU to a GPLNL request is shown in table 218.

Table 218 – Accept CT_IU to GPLNL Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Number of Platform Node Name entries (n)	4
Platform Node Name #1	8
Platform Node Name #2	8
...	
Platform Node Name #n	8

One or more Platform Node Name entries are returned, and the entries may be returned in any order, and the order may be different for every request even if the same entries are returned and the requestor is the same.

6.2.5.24 Query - Get Platform Type (GPLT)

The Fabric Configuration Server shall, if it receives a GPLT request, return the Platform Type attribute for the specified Platform Name. The format of the GPLT Request CT_IU is shown in table 219.

Table 219 – GPLT Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Platform Name	256

If the GPLT request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the Accept CT_IU to a GPLT request is shown in table 220.

Table 220 – Accept CT_IU to GPLT Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Platform Type	4

6.2.5.25 Query - Get Platform Management Address List (GPLML)

The Fabric Configuration Server shall, if it receives a GPLML request, return all Interconnect Element Management Address attributes for the specified Platform Name. The format of the GPLML Request CT_IU is shown in table 221.

Table 221 – GPLML Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Platform Name	256

If the GPLML request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the Accept CT_IU to a GPLML request is shown in table 222.

Table 222 – Accept CT_IU to GPLML Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Number of Management Address entries (n)	4
Management Address #1	256
Management Address #2	256
...	
Management Address #n	256

One or more Platform Management Address entries are returned, and the entries may be returned in any order, and the order may be different for every request even if the same entries are returned and the requestor is the same.

6.2.5.26 Query - Get Platform Attribute Block (GPAB)

The Fabric Configuration Server shall, if it receives a GPAB request, return the Platform Attribute Block for the specified Platform. The GPAB request payload shall specify the Platform Name that identifies the Platform for which attributes are sought. The GPAB accept payload contains the Platform Attribute Block for the Platform specified in the request.

If the specified Platform has not been registered with the Fabric Configuration Server, then the request is rejected with the appropriate reason code explanation.

If the GPAB request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the GPAB Request payload is depicted in table 223.

Table 223 – GPAB Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Platform Name	256

Platform Name: The format of the Platform Name is described in 6.2.3.4.2.

The format of the GPAB Accept payload is depicted in table 224.

Table 224 – GPAB Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Platform Attribute Block	see 6.2.3.4.4

Platform Attribute Block: The format of the Platform Attribute block is described in 6.2.3.4.4.

6.2.5.27 Query - Get Platform Name - Node Name (GNPL)

The Fabric Configuration Server shall, if it receives a GNPL request, return the Platform Name attribute for the specified Platform Node Name. The format of the GNPL Request CT_IU is shown in table 225.

Table 225 – GNPL Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Platform Node Name	8

If the GNPL request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the Accept CT_IU to a GNPL request is shown in table 226.

Table 226 – Accept CT_IU to GNPL Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Platform Name	256

6.2.5.28 Query - Get Platform Name List (GPNL)

The Fabric Configuration Server shall, if it receives a GPNL request, return a list of all registered Platform Names. If no Platform Names are Registered, an GPNL Accept CT_IU shall be sent with the Number of Platform Name entries field set to zero, and with no Platform Names in the payload. The format of the GPNL Request CT_IU is shown in table 227.

Table 227 – GPNL Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

If the GPNL request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the Accept CT_IU to a GPNL request is shown in table 228.

Table 228 – Accept CT_IU to GPNL Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Number of Platform Name entries (n)	4
Platform Name #1	256
Platform Name #2	256
...	
Platform Name #n	256

6.2.5.29 Query - Get Platform FCP Type (GPFCP)

The Fabric Configuration Server shall, if it receives a GPFCP request, return the FCP-4 Feature Bit Mask (see 6.2.3.4.8.2) for the specified Platform Name. The format of the GPFCP Request CT_IU is shown in table 229.

Table 229 – GPFCP Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Platform Name	256

If the GPFCP request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the Accept CT_IU to a GPFCP request is shown in table 230.

Table 230 – Accept CT_IU to GPFCP Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
FCP-4 Features Bit Mask	4

6.2.5.30 Query - Get Platform OS LUN Mappings (GPLI)

The Fabric Configuration Server shall, if it receives a GPLI request, return all OS LUN Map entry attributes for the specified Platform Name. The format of the GPLI Request CT_IU is shown in table 231.

Table 231 – GPLI Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Platform Name	256

If the GPLI request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the Accept CT_IU to a GPLI request is shown in table 232.

Table 232 – Accept CT_IU to GPLI Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Number of LUN Map entries (n)	4
OS LUN Map Entry #1	variable
OS LUN Map Entry #2	variable
...	
OS LUN Map Entry #n	variable

One or more OS LUN Map entries are returned, and the entries may be returned in any order, and the order may be different for every request even if the same entries are returned and the requestor is the same.

The format of the OS LUN Map entries is shown in table 167.

6.2.5.31 Query - Get Node Identification Data (GNID)

The Fabric Configuration Server shall, if it receives a GNID request, return the Topology Information for the specified destination. The format of the GNID Request CT_IU is shown in table 233.

Table 233 – GNID Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Platform Node Name	8
Node Identification Data format	1
Reserved	3

If the GNID request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

See the Request Node Identification Data ELS in FC-FS-4 for the definition of the Node Identification Data format.

The format of the Accept CT_IU to a GNID request is shown in table 234.

Table 234 – Accept CT_IU to GNID Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
RNID accept payload	see text

The RNID accept payload shall contain the accept payload for the Request Node Identification Data ELS. See FC-FS-4 for the definition and length of this payload.

6.2.5.32 Register Interconnect Element Logical Name (RIELN)

The RIELN Fabric Configuration Server request shall be used to associate a Logical Name with a given Interconnect Element.

The Fabric Configuration Server shall not attempt validation of the Logical Name attribute. This means that any Logical Name value shall be accepted.

Deregistration may be accomplished by registering a null Logical Name (see 6.2.3.2.6).

If the RIELN request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the RIELN Request CT_IU is shown in table 235.

Table 235 – RIELN Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Interconnect Element Name	8
Logical Name	256

The format of the RIELN Accept CT_IU is shown in table 236.

Table 236 – RIELN Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.2.5.33 Register Platform (RPL)

The RPL Fabric Configuration Server request shall be used to associate a Platform Name with its Platform Node Names and key Platform attributes. This request allows the registration of a Platform, associated Platform Node Names, and key attributes using a single transaction.

The Platform Name field of the Request CT_IU shall not be equal to a currently registered Platform Name.

If the value of the Platform Name is equal to a currently registered Platform Name maintained by the Switch to which the Request CT_IU is addressed, the Fabric Configuration Server of the Switch responding to the CT_IU request shall reject this request; the Reject CT_IU reason code shall be “Unable to perform command request”, with a reason code explanation of “Platform Name already exists”.

If the device (e.g., Switch) from which the Request CT_IU is sent is part of the Fabric Configuration Server and the Request CT_IU is rejected the sending device shall deregister the Platform specified in the Request CT_IU.

No Platform Node Name field of the Request CT_IU shall be equal to a currently registered Platform Node Name. If the value of any Platform Node Name is equal to a currently registered Platform Node Name, the Fabric Configuration Server shall reject this request; the Reject CT_IU reason code shall be ‘Unable to perform command request’, with a reason code explanation of ‘Platform Node Name already exists’

In the absence of a reject condition, the Fabric Configuration Server shall create a new Platform object and register the Platform attributes with the new Platform.

The Fabric Configuration Server shall not attempt validation of the Platform Type or Platform Management Address attributes. This means that any value shall be accepted for these attributes.

Deregistration may be accomplished by a DPL request (see 6.2.5.40).

If the RPL request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the RPL Request CT_IU is shown in table 237.

Table 237 – RPL Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Platform Name	256
Platform Type	4
Number of Management Address entries (n)	4
Management Address #1	256
Management Address #2	256
...	
Management Address #n	256
Number of Platform Node Name entries (n)	4
Platform Node Name #1	8
Platform Node Name #2	8
...	
Platform Node Name #n	8

The format of the RPL Accept CT_IU is shown in table 238.

Table 238 – RPL Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.2.5.34 Register Platform Node Name (RPLN)

The RPLN Fabric Configuration Server request shall be used to associate a Platform Node Name with a Platform.

The Platform Name field of the Request CT_IU may be equal to a currently registered Platform Name. If the value of the Platform Name is equal to a currently registered Platform Name, the Fabric Configuration Server shall register the Platform Node Name with the existing Platform. If the value of the Platform Name is not equal to a currently registered Platform Name, the Fabric Configuration Server shall create a new Platform object and register the Platform Node Name with the new Platform.

No Platform Node Name field of the Request CT_IU shall be equal to a currently registered Platform Node Name. If the value of any Platform Node Name is equal to a currently registered Platform Node

Name, the Fabric Configuration Server shall reject this request; the Reject CT_IU reason code shall be ‘Unable to perform command request’, with a reason code explanation of ‘Platform Node Name already exists’.

The Fabric Configuration Server shall reject this request if the Platform Node Name attribute is currently assigned to another Platform.

Deregistration may be accomplished by a DPLN request (see 6.2.5.41).

If the RPLN request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the RPLN Request CT_IU is shown in table 239.

Table 239 – RPLN Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Platform Name	256
Platform Node Name	8

The format of the RPLN Accept CT_IU is shown in table 240.

Table 240 – RPLN Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.2.5.35 Register Platform Type (RPLT)

The RPLT Fabric Configuration Server request shall be used to associate a Platform Type with a Platform.

The Platform Name field of the Request CT_IU may be equal to a currently registered Platform Name. If the value of the Platform Name is equal to a currently registered Platform Name, the Fabric Configuration Server shall register the Platform Type with the existing Platform. If the value of the Platform Name is not equal to a currently registered Platform Name, the Fabric Configuration Server shall create a new Platform object and register the Platform Type with the new Platform.

The Fabric Configuration Server shall not attempt validation of the Platform Type attribute. This means that any value shall be accepted for this attribute.

Deregistration may be accomplished by registering a null Platform Type (see 6.2.3.4.3).

If the RPLT request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the RPLT Request CT_IU is shown in table 241.

Table 241 – RPLT Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Platform Name	256
Platform Type	4

The format of the RPLT Accept CT_IU is shown in table 242.

Table 242 – RPLT Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.2.5.36 Register Platform Management Address (RPLM)

The RPLM Fabric Configuration Server request shall be used to associate a Platform Management Address with a Platform.

The Platform Name field of the Request CT_IU may be equal to a currently registered Platform Name. If the value of the Platform Name is equal to a currently registered Platform Name, the Fabric Configuration Server shall register the Platform Management Address with the existing Platform. If the value of the Platform Name is not equal to a currently registered Platform Name, the Fabric Configuration Server shall create a new Platform object and register the Platform Management Address with the new Platform.

The Fabric Configuration Server shall not attempt validation of the Platform Management Address attribute. This means that any value shall be accepted for this attribute.

Deregistration may be accomplished by a DPLML request (see 6.2.5.43).

If the RPLM request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the RPLM Request CT_IU is shown in table 243.

Table 243 – RPLM Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Platform Name	256
Platform Management Address	256

The format of the RPLM Accept CT_IU is shown in table 244.

Table 244 – RPLM Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.2.5.37 Register Platform Attribute Block (RPAB)

The Fabric Configuration Server shall if it receives an RPAB request, register the Platform attributes for the specified Platform. The RPAB request payload shall specify the Platform Name and the Platform Attribute Block.

The Platform Name field of the Request CT_IU may be equal to a currently registered Platform Name. If the value of the Platform Name is equal to a currently registered Platform Name, the Fabric Configuration Server shall register the Platform Attribute Block with the existing Platform. If the value of the Platform Name is not equal to a currently registered Platform Name, the Fabric Configuration Server shall reject this request. The Reject CT_IU reason code shall be ‘Unable to perform command request’, with a reason code explanation of ‘Platform name does not exist’.

The Vendor ID and Product ID attributes shall be required in the Platform Attribute Block. If the Vendor ID or the Product ID attribute are not specified in the Platform Attribute Block then the request is failed with the appropriate reason code explanation.

If a Platform Attribute Block is already registered for the Platform indicated in the Platform Name field then that Platform Attribute Block shall replace the existing Platform Attribute Block.

If the RPAB request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the RPAB Request payload is depicted in table 245.

Table 245 – RPAB Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Platform Name	256
Platform Attribute Block	see 6.2.3.4.4

Platform Name: The format of the Platform Name is described in 6.2.3.4.2.

Platform Attribute Block: The format of the Platform Attribute block is described in 6.2.3.4.4.

The format of the RPAB Accept payload is depicted in table 246.

Table 246 – RPAB Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.2.5.38 Register Platform FCP Type (RPFCP)

The RPFCP Fabric Configuration Server request shall be used to associate a Platform FCP Type with a Platform.

The Platform Name field of the Request CT_IU may be equal to a currently registered Platform Name. If the value of the Platform Name is equal to a currently registered Platform Name, the Fabric Configuration Server shall register the Platform FCP Type with the existing Platform. If the value of the Platform Name is not equal to a currently registered Platform Name, the Fabric Configuration Server shall reject this request. The Reject CT_IU reason code shall be ‘Unable to perform command request’, with a reason code explanation of ‘Platform name does not exist’.

The Fabric Configuration Server shall not attempt validation of the Platform FCP Type attribute. This means that any value shall be accepted for this attribute.

Deregistration may be accomplished by registering a null Platform FCP Type (see 6.2.3.4.8.2).

If the RPFCP request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the RPFCP Request CT_IU is shown in table 247.

Table 247 – RPFCP Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Platform Name	256
Platform FCP Type	4

The format of the RPFCP Accept CT_IU is shown in table 248.

Table 248 – RPFCP Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.2.5.39 Register Platform OS LUN Mappings (RPLI)

The RPLI Fabric Configuration Server request shall be used to associate a Platform Name with one or more OS LUN mappings.

The Platform Name field of the Request CT_IU may be equal to a currently registered Platform Name. If the value of the Platform Name is equal to a currently registered Platform Name, the Fabric Configuration Server shall register the Platform OS LUN mapping(s) with the existing Platform. If the value of the Platform Name is not equal to a currently registered Platform Name, the Fabric Configuration Server shall reject this request. The Reject CT_IU reason code shall be ‘Unable to perform command request’, with a reason code explanation of ‘Platform name does not exist’.

If an OS LUN Map Entry is already registered for the Platform indicated in the Platform Name field then the requested OS LUN Mapping(s) shall add to the existing OS LUN mapping. Checking for duplicate OS LUN Map Entry attributes is not required by the Fabric Configuration Server.

The Fabric Configuration Server shall reject this request if the number of LUN map entries is set to zero. The Reject CT_IU reason code shall be ‘Unable to perform command request’, with a reason code explanation of ‘Zero entries in OS LUN map block’.

Deregistration may be accomplished by a DPLI request (see 6.2.5.44).

If the RPLI request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the RPLI Request CT_IU is shown in table 249.

Table 249 – RPLI Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Number of LUN Map entries (n)	4
OS LUN Map Entry #1	variable
OS LUN Map Entry #2	variable
...	
OS LUN Map Entry #n	variable

One or more OS LUN Map entries may be sent.

The format of the OS LUN Map entries is shown in table 167.

The format of the RPL Accept CT_IU is shown in table 250.

Table 250 – RPLI Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.2.5.40 Deregister Platform (DPL)

The DPL Fabric Configuration Server request shall be used to remove a registered Platform object and all of its attributes.

The Platform Name field of the Request CT_IU shall be equal to a currently registered Platform Name. If the value of the Platform Name is not equal to a currently registered Platform Name, the Fabric Configuration Server shall reject this request; the Reject CT_IU reason code shall be ‘Unable to perform command request’, with a reason code explanation of ‘Platform Name does not exist’

If the value of the Platform Name is equal to a currently registered Platform Name, the Fabric Configuration Server shall delete the Platform object from its database.

If the DPL request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the DPL Request CT_IU is shown in table 251.

Table 251 – DPL Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Platform Name	256

The format of the DPL Accept CT_IU is shown in table 252.

Table 252 – DPL Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.2.5.41 Deregister Platform Node Name (DPLN)

The DPLN Fabric Configuration Server request shall be used to disassociate a registered Platform Node Name from a Platform object.

The Platform Node Name field of the Request CT_IU shall be equal to a currently registered Platform Node Name. If the value of the Platform Node Name is not equal to a currently registered Platform Node Name, the Fabric Configuration Server shall reject this request; the Reject CT_IU reason code shall be ‘Unable to perform command request’, with a reason code explanation of ‘Platform Node Name does not exist’.

If the value of the Platform Node Name is equal to a currently registered Platform Node Name, the Fabric Configuration Server shall delete the Platform Node Name from its database.

If the DPLN request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the DPLN Request CT_IU is shown in table 253.

Table 253 – DPLN Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Platform Node Name	8

The format of the DPLN Accept CT_IU is shown in table 254.

Table 254 – DPLN Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.2.5.42 Deregister Platform Management Address (DPLM)

The Configuration Server shall, if it receives a DPLM request, delete the specified Platform Management Address for the specified Platform. The DPLM request payload shall contain the Platform Name and a Platform Management Address (see table 255). The DPLM accept payload is null (see table 256).

Table 255 – DPLM Request Payload

Item	Size (Bytes)
CT_IU Preamble	see 4.3
Platform Name	256
Platform Management Address	256

If the DPLM request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

Table 256 – DPLM Accept Payload

Item	Size (Bytes)
CT_IU Preamble	see 4.3

6.2.5.43 Deregister Platform Management Address List (DPLML)

The DPLML Fabric Configuration Server request shall be used to disassociate all registered Platform Management Addresses from a Platform object.

The Platform Name field of the Request CT_IU shall be equal to a currently registered Platform Name. If the value of the Platform Name is not equal to a currently registered Platform Name, the Fabric Configuration Server shall reject this request; the Reject CT_IU reason code shall be ‘Unable to perform command request’, with a reason code explanation of ‘Platform Name does not exist’

If the value of the Platform Name is equal to a currently registered Platform Name, the Fabric Configuration Server shall delete all Platform Management Addresses, associated with the Platform object, from its database.

If the DPLML request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the DPLML Request CT_IU is shown in table 257.

Table 257 – DPLML Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Platform Name	256

The format of the DPLML Accept CT_IU is shown in table 258.

Table 258 – DPLML Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.2.5.44 Deregister Platform OS LUN Mappings (DPLI)

The DPLI Fabric Configuration Server request shall be used to delete all the OS LUN Map Entry objects and attributes associated with the indicated Platform.

The Platform Name field of the Request CT_IU shall be equal to a currently registered Platform Name. If the value of the Platform Name is not equal to a currently registered Platform Name, the Fabric Configuration Server shall reject this request; the Reject CT_IU reason code shall be ‘Unable to perform command request’, with a reason code explanation of ‘Platform Name does not exist’.

If the indicated Platform does not contain any OS LUN Entry objects, the Fabric Configuration Server shall reject this request; the Reject CT_IU reason code shall be ‘Unable to perform command request’, with a reason code explanation of ‘Zero entries in OS LUN Map’.

If the value of the Platform Name is equal to a currently registered Platform Name, the Fabric Configuration Server shall delete all the OS LUN Map Entry objects for the indicated Platform from its database.

If the DPLI request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the DPLI Request CT_IU is shown in table 259.

Table 259 – DPLI Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Platform Name	256

The format of the DPLI Accept CT_IU is shown in table 260.

Table 260 – DPLI Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.2.5.45 Deregister Platform Attribute Block (DPAB)

The Fabric Configuration Server shall, if it receives an DPAB request, de-register the Platform Attribute Block for the specified Platform. The DPAB request payload shall specify the Platform Name that identifies the Platform for which the Platform Attribute Block is to be de-registered.

If the specified Platform or its attribute block have not been registered with the Fabric Configuration Server, then the request is rejected with the appropriate reason code explanations.

If the DPAB request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the DPAB Request payload is depicted in table 261.

Table 261 – DPAB Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Platform Name	256

The format of the DPAB Accept payload is depicted in table 262 below.

Table 262 – DPAB Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.2.5.46 De-Register All Platform Information (DPALL)

The Fabric Configuration Server shall, if it receives a DPALL operation request, delete all Platforms and their attributes from the Platform database. The format of the DPALL request payload shall be as shown in table 263.

Table 263 – DPALL Request Payload

Item	Size (Bytes)
CT_IU Preamble	see 4.3

If the DPALL request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

The format of the DPALL accept payload shall be as shown in table 264.

Table 264 – DPALL Accept Payload

Item	Size (Bytes)
CT_IU Preamble	see 4.3

6.2.5.47 FC Trace Route (FTR)

The FTR request obtains the path information between two Fx_Ports from the Fabric Configuration Server. The path information includes the Switch_Name, Domain_ID, Port_Name and Physical Port Number for Switch Ports that route traffic between the two Fx_Ports (e.g., Source Port and Destination Port). The path information shall be collected for the Fabric ingress path (from Source Port to Destination Port) and for the Fabric egress path (from Destination Port to Source Port).

Figure 9 shows a four-Switch Fabric that is used for a FC Trace Route example. When a Management Application on Device C requests a Trace Route between Devices A and B (Source and Destination Port), Switch 4 contains the Fabric Configuration Server and shall send a Switch Trace Route (STR) SW_ILS to the Source Port Switch's (Switch 1) Domain Controller ID. Switch 1 shall start the STR processing that is shown in figure 10.

Each Switch in the ingress path shall accept the STR, append its ingress Path Information, increment the number of Path Information Entries, and send an STR Request to the next Switch in the ingress path. This process continues until the STR Request reaches the Destination Port's Switch (Switch 4). After receiving the STR Request, Switch 4 shall append its ingress and egress Path Information. The Destination Port's Switch shall increment the number of Path Information Entries by two, and send an STR Request to the next Switch in the egress path. Each intermediate Switch shall process the STR Requests until the request reaches the Source Port's Switch in the egress Path.

After the Source Port's Switch completes processing the STR request, the final STR request is sent to the Managing Switch. The Managing Switch accepts the STR, converts the information in the STR Request to the FTR Response, and sends the FTR response to the FTR Requestor.

If the Fabric Configuration Server is not able to answer the FTR Request within three times R_A_TOV, it shall use GS Asynchronous Notification and reject the FTR Request with a Reject CT_IU reason code of "Unable to perform command request" and a reason code explanation of "Processing Request".

After rejecting the request and receiving the completed Switch Trace Route SW_ILS, the Fabric Configuration Server shall use GS Asynchronous Notification to notify the requesting Nx_Port that the Accept CT_IU for the FTR Request is ready.

If the two Nx_Ports are not in a common Zone, the FTR Request shall be rejected with the Reject CT_IU reason code of “Logical Error” and a Reject CT_IU reason code explanation of “Devices not in a common Zone”.

If the FTR request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

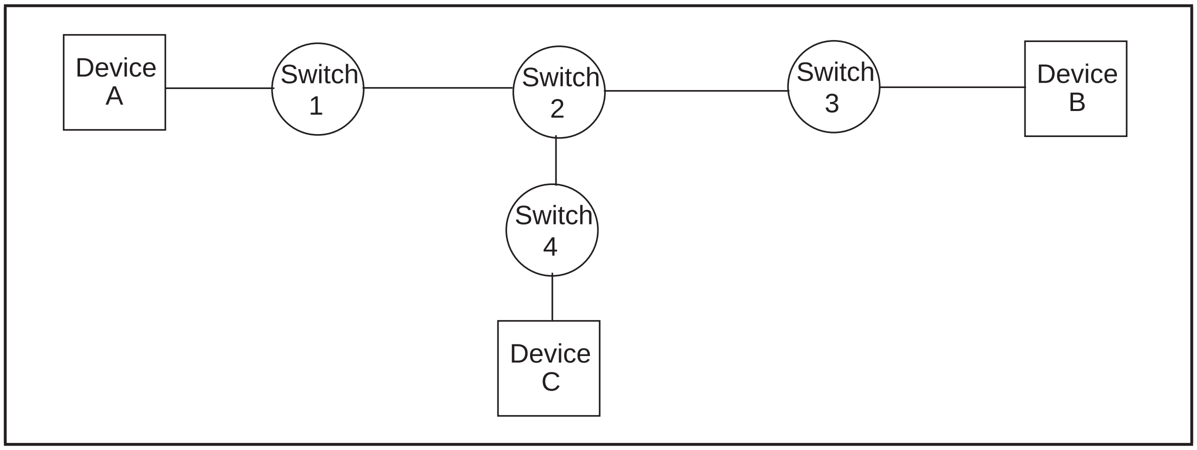


Figure 9 – Topology for FTR Example.

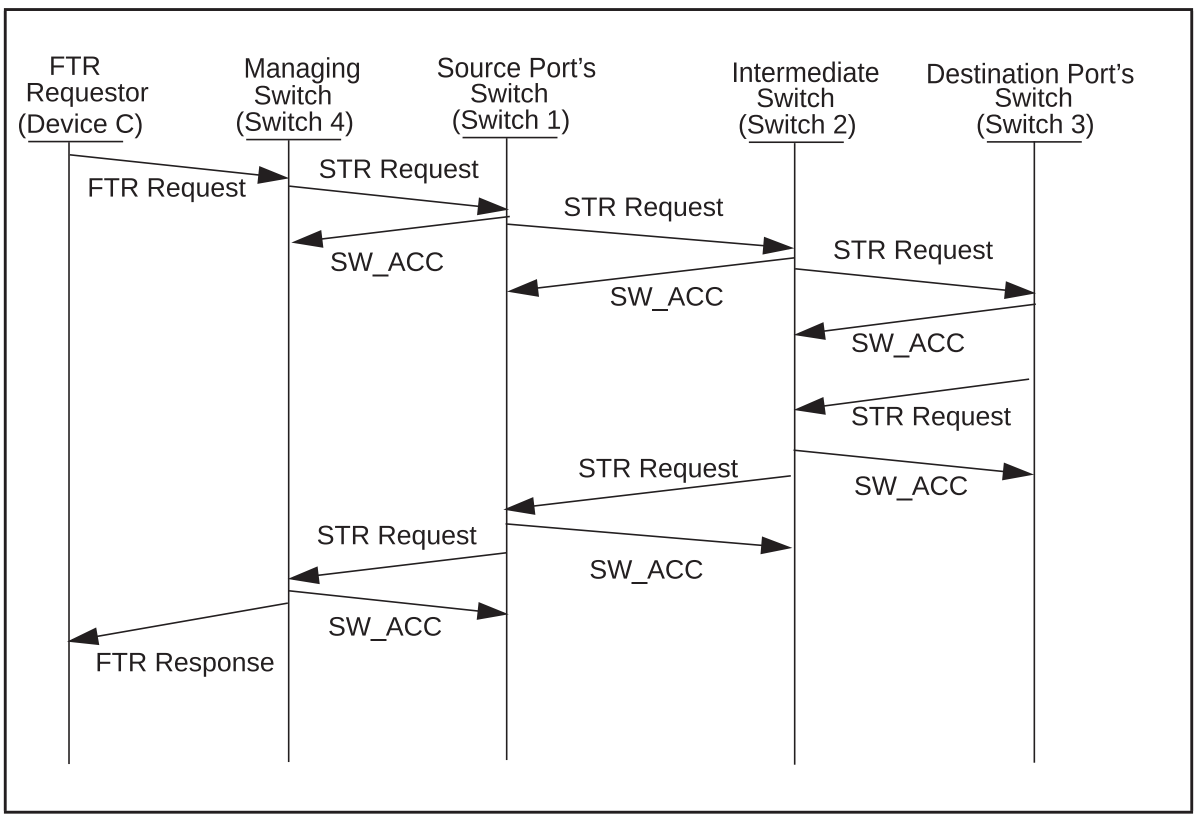


Figure 10 – FTR Processing

The Fabric Configuration Server may use GS Asynchronous Notification during the processing of the FTR request. The format of the FTR request is shown in table 265.

Table 265 – FTR Request CT_IU

Item	Size (Bytes)
CT_IU Preamble	see 4.3
Revision	4
Source Port Tag	2
Source Port Length	2
Source Port Value	n
Destination Port Tag	2
Destination Port Length	2
Destination Port Value	n
Token	4
T10 Vendor Identifier	8
Vendor Specific Information	8
Maximum Hop Count	4

Revision: The revision shall contain a value of 01h.

Source Port Tag: The tag shown in table 266 identifies the Source Port format.

Table 266 – Source/Destination Port Tags

Tag	Item
01h	N_Port_ID
02h	N_Port_Name
Others	reserved

Source Port Length: The length of the Source Port Value in bytes. The length shall be a multiple of four.

Source Port Value: The value of the Source Port defined by the Source Port Tag. Fill bytes are added as necessary to the end of the actual value in order to ensure that the length of the value field is a multiple of four. Fill bytes shall be nulls (00h). The number of fill bytes (f) is zero, one, two or three depending on the length of the actual value (m). The total length of the value field is (n = f + m).

Destination Port Tag: The tag used to identify the Destination Port format as shown in Table 266.

Destination Port Length: The length of the Destination Port Value in bytes. The length shall be a multiple of four.

Destination Port Value: The value of the Destination Port defined by the Destination Port Tag. Fill bytes are added as necessary to the end of the actual value in order to ensure that the length of the value field is a multiple of four. Fill bytes shall be nulls (00h). The number of fill bytes (f) is zero, one, two or three depending on the length of the actual value (m). The total length of the value field is (n = f + m).

Token: An identifier provided by the requesting Nx_Port. The token shall be incremented for each new request. Tokens shall be 32-bit unsigned integers that shall wrap to zero on exceeding 7FFF FFFFh. A token value of FFFF FFFFh shall indicate the token value is NULL. If an identical token is already being processed, the FTR Request shall be rejected with a CT_IU reason code of “Logical Error” with a Reject CT_IU reason code explanation of “Command already in Progress”.

T10 Vendor Identifier: The T10 Vendor Identifier field shall contain the ~~vendor s~~ eight byte T10 administered vendor identifier of the vendor that defines the content of the Vendor Specific field. A Null value consists of eight space characters. If the T10 Vendor ID contains a Null value, the Vendor Specific field shall be ignored.

Vendor Specific Information: The Vendor Specific Information field shall contain the vendor’s information. The format of the information is defined by the vendor and not by this standard.

Maximum Hop Count: The maximum number of hops in the roundtrip path between the Source Port and the Destination Port. The default value for the Maximum Hop Count is 20.

After receiving the FTR Request, the Fabric Configuration Server may accept the request with the payload specified in table 267.

Table 267 – Accept CT_IU to FTR Request

Item	Size (Bytes)
CT_IU Preamble	see 4.3
00000001h	4
Token	4
T10 Vendor Identifier	8
Vendor Specific Information	8
STR Reject Reason Code	4
Number of Path Information Entries	4
Source Port’s Fabric Ingress Path Information	36
Intermediate Switch’s Path Information	36
...	36
Destination Port’s Fabric Ingress Path Information	36
Destination Port’s Fabric Egress Path Information	36
...	36
Source Port’s Fabric Egress Path Information	36

Revision: The revision shall contain a value of 01h.

Token: An identifier provided by the requesting Nx_Port.

T10 Vendor Identifier: The T10 Vendor Identifier field shall contain the vendor’s eight byte T10 administered vendor identifier.

Vendor Specific Information: The Vendor Specific Information field shall contain the vendor’s information. The format of the information is defined by the vendor and not by this standard.

STR Reject Reason Code: If the command was not completed successfully, the Switch Trace Route Command FC Trace Route shall include a reject reason code that is defined in FC-SW-6. If the command is completed successfully, the value is 00 00 00 00h.

Number of Path Information Entries: The number of Path Information Entries. This number shall be incremented by one for each Path Information entry that a Switch appends to the Trace Route SW_ILS.

Source Port’s Fabric Ingress Path Information: The Path Information for the Source Port in the Fabric ingress path. The format of the Path Information is shown in Table 268.

Table 268 – Path Information

Item	Size (Bytes)
Switch Name	8
Domain_ID	4
Ingress Port_Name	8
Ingress Physical Port Number	4
Egress Port_Name	8
Egress Physical Port Number	4

Switch Name: The Switch Name of the Switch in the path that is appending the path information.

Domain_ID: The Domain_ID of the Switch reporting the Path Information. The format of Domain_ID shall be set to 000000h||Domain_ID’

Ingress Port_Name: The Port_Name of the F_Port or E_Port on the Switch that the frame enters.

Ingress Physical_Port_Number: The Physical_Port_Number of the F_Port or E_Port on the Switch that the frame enters.

Egress Port_Name: The Port_Name of the F_Port or E_Port on the Switch that the frame exits.

Egress Physical_Port_Number: The Physical_Port_Number of the F_Port or E_Port on the Switch that the frame exits.

Intermediate Switch’s Path Information: The Path Information (see table 268) for an intermediate Switch in the Fabric ingress path, if any.

Destination Port’s Ingress Path Information: The Path Information (see table 268) Destination Port in the Fabric ingress path.

Destination Port’s Egress Path Information: The Path Information (see table 268) for the Destination Port in the Fabric egress path.

Source Port’s Egress Path Information: The Path Information (see table 268) for the Source Port in the Fabric egress path.

6.2.5.48 FC Ping (FPNG)

The FC Ping request verifies the path to an Nx_Port or Domain Controller is functional through the use of the Echo ELS (see FC-LS-3). The Entry Switch shall send the Echo ELS to the Nx_Port to confirm that the connection is functioning properly. Having the Entry Switch send the Echo ELS bypasses Zoning restrictions that might limit a management station from sending an Echo to the Nx_Port directly.

Figure 11 and figure 12 illustrate the process of pinging an Nx_Port in the Fabric. When a Management Application on device C sends a FC Ping request to the Fabric Configuration Server for

Device A, the Entry Switch (Switch 4) shall send an Echo ELS to the Nx_Port. After Switch 4 receives the ECHO response from the Nx_Port, Switch 4 accepts the FC Ping Request.

If the Echo ELS does not receive a response, the Fabric Configuration Server shall reject the FC Ping Request with a Reject CT_IU reason code of “Unable to perform command request” and a Reject CT_IU reason code explanation of “Unable to Verify Connection”.

If the FPNG request is not supported a Reject CT_IU should be returned and the reason code should be Command Not Supported, no further explanation.

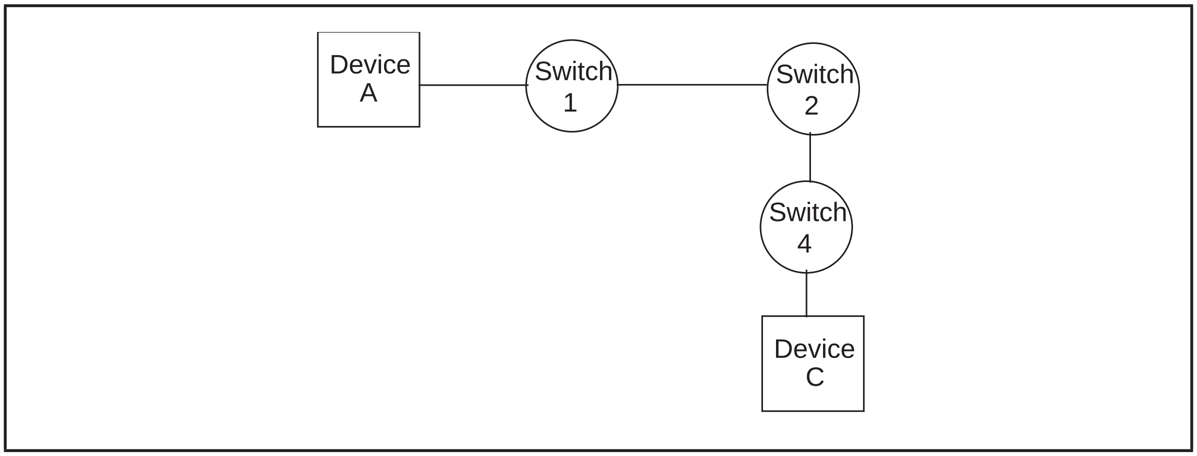


Figure 11 – Topology for FC Ping Example.

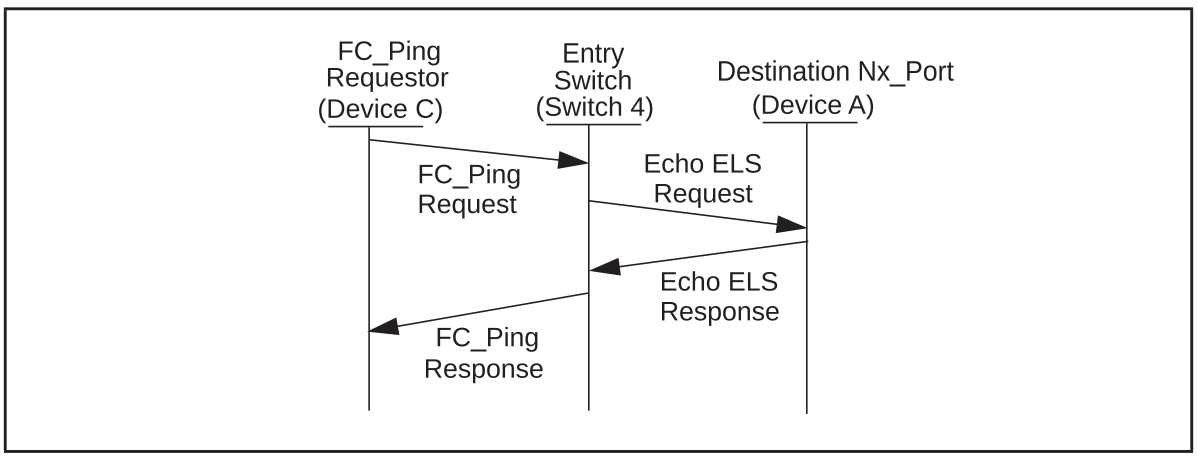


Figure 12 – FC Ping Processing

The format of the FC Ping Request is shown in table 269.

Table 269 – FC_Ping Request CT_IU

Item	Size (Bytes)
CT_IU Preamble	see 4.3
00000001h	4
Port Tag	2
Port Length	2
Port Value	n
Token	4

Revision: The revision shall contain a value of 01h.

Port Tag: The tag shown in table 270 that identifies the Port format.

Table 270 – Nx_Port Tags

Tag	Item
01h	N_Port_ID
02h	Port_Name

Port Length: The length of the Port Value in bytes. The length shall be a multiple of four.

Port Value: The value of Port defined by the Port Tag. Fill bytes are added as necessary to the end of the actual value in order to ensure that the length of the value field is a multiple of four. Fill bytes shall be nulls (00h). The number of fill bytes (f) is zero, one, two or three depending on the length of the actual value (m). The total length of the value field is (n = f + m).

Token: An identifier provided by the requesting Nx_Port. The token shall be incremented for each new request. Tokens shall be 32-bit unsigned integers that shall wrap to zero on exceeding 7FFF FFFFh. A token value of FFFF FFFFh shall indicate the token value is NULL. If an identical token is already being processed, the FC Ping Request shall be rejected with a Reject CT_IU reason code of “Logical Error” with a Reject CT_IU reason code explanation of “Processing Request”.

After receiving the FC Ping Request, the Fabric Configuration Server may accept the request with the payload specified in table 271.

Table 271 – Accept CT_IU to FC Ping Request

Item	Size (Bytes)
CT_IU Preamble	(see 4.3)
Token	4

Token: An identifier provided by the requesting Nx_Port.

6.3 Unzoned Name Server

6.3.1 Overview

The Unzoned Name Server is provided by the Management Service to give a management application access to the Name Server database without Zone constraints.

6.3.2 Protocol

6.3.2.1 Overview

Unzoned Name Server registration, deregistration and queries are managed through protocols containing a set of Request CT_IUs and Response CT_IUs supported by the Unzoned Name Server.

For a Unzoned Name Server request, the Unzoned Name Server payload shall be transported from the requestor to the Unzoned Name Server using a Request CT_IU. The corresponding Unzoned Name Server response is transported from the Unzoned Name Server to the requestor, in the Exchange established by the requestor, using a Response CT_IU.

If Zones exist within the Fabric, the Unzoned Name Server shall not restrict access to information in the Name Server database based on the Zone configuration.

An example of Zoned and Unzoned access is illustrated in figure 13.

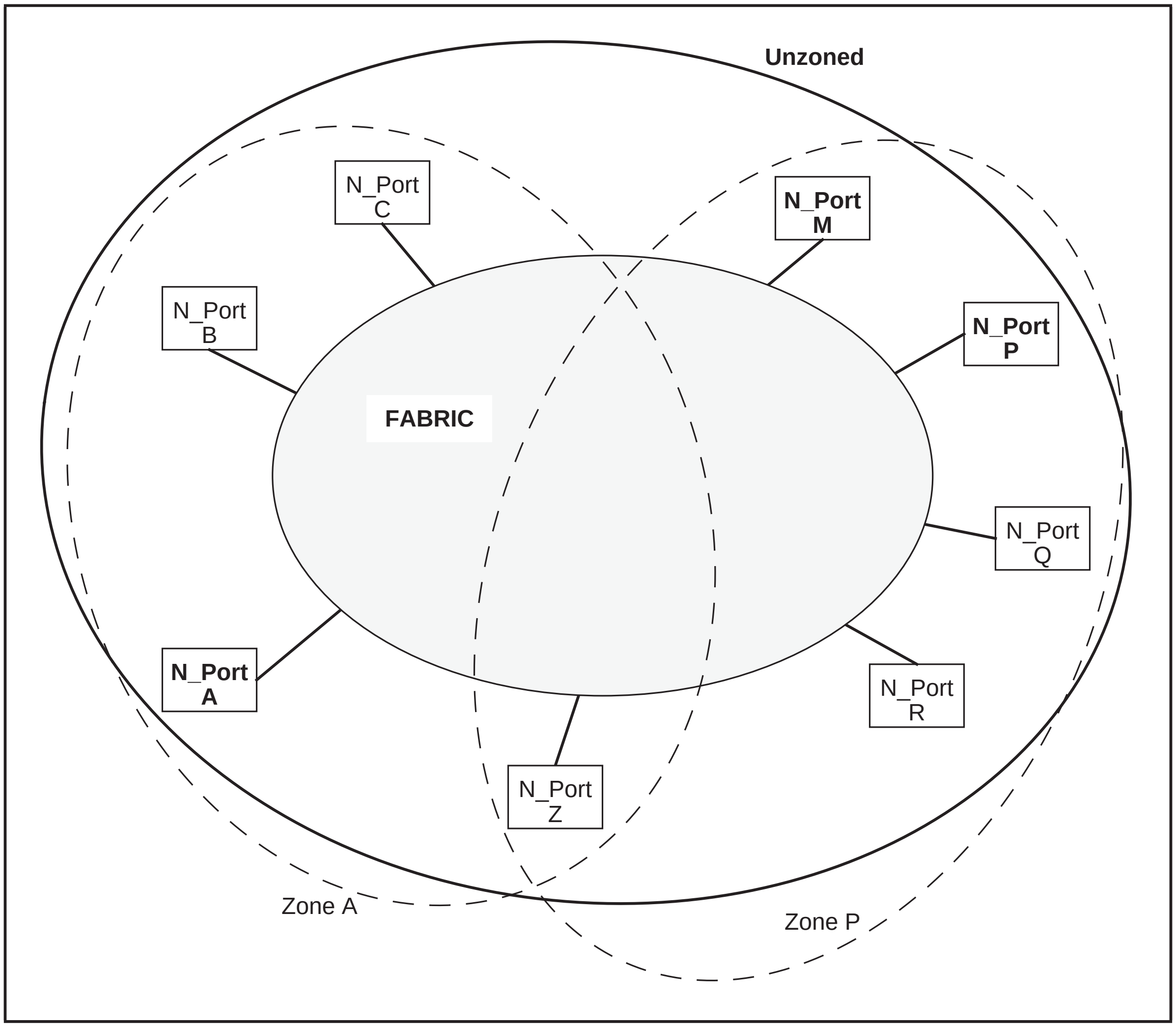


Figure 13 – Name Server Zone Constraints

In the example, N_Port A is in a Zone that allows access to, and visibility of, N_Ports B, C, and Z. N_Port P is in a Zone that allows access to, and visibility of, N_Ports Q, R, M, and Z. N_Port M is in the same Zone as P, and is allowed access to, and visibility of, N_Ports Q, R, P, and Z.

The Name Server provided by the Directory Service is required to constrain access based on Zones (see 5.2.2). So, if each N_Port issues a GID_PT (Get IDs based on Port Type) request, the answers they get are:

a) Response for N_Port A contains N_Ports B, C, and Z;

b) Response for N_Port P contains N_Ports Q, R, M, and Z; and

c) Response for N_Port M contains N_Ports P, Q, R, and Z.

If that N_Port M is running a management application, and makes the same GID_PT request via the Unzoned Name Server provided by the Management Service. The response for N_Port M in this case contains N_Ports A, B, C, P, Q, R, M, and Z. The response is unconstrained by the Zone configuration.

6.3.2.2 CT_IU preamble values

The following values shall be set in the CT_IU preamble for Unzoned Name Server request and their responses; fields not specified here shall be set as defined in 4.3.2:

a) GS_Subtype: as indicated in table 125; and

b) Command Code: see table 22 for Request command codes.

6.3.2.3 Registration

The Unzoned Name Server shall not perform Registrations. If the Unzoned Name Server receives any Registration command defined in 5.2.2.3, it shall reject the command. The Reject CT_IU reason code shall be ‘Unable to perform command request’, with a reason code explanation of ‘Access denied’.

6.3.2.4 Queries

Queries shall be performed as defined in 5.2.2.4.

6.3.2.5 Deregistration

The Unzoned Name Server shall not perform Deregistration. If the Unzoned Name Server receives any Deregistration command defined in 5.2.2.5, it shall reject the command. The Reject CT_IU reason code shall be ‘Unable to perform command request’, with a reason code explanation of ‘Access denied’.

6.3.2.6 Server Sessions

The action of the Unzoned Name Server is unaffected by Server Sessions.

6.3.3 Unzoned Name Server objects - Formats

Unzoned Name Server objects are as defined in 5.2.3.

6.3.4 Reason code explanations

Unzoned Name Server reason code explanations are as defined in 5.2.4.

6.3.5 Commands

The commands defined for the Unzoned Name Server are summarized in table 22, and are defined in 5.2.5.

6.4 Fabric Zone Server

6.4.1 Overview

Fabric Zones provide a mechanism to expose selected views of Name Server information to Clients or control frame delivery between Nx_Ports. This technique is similar to “virtual private networks” in that the Fabric has the ability to group devices into Zones. The following discussion provides a brief architectural overview of Zones, describes the parameters that define a Zone, and describes the relationships, and protocols for managing Zone information and configurations.

Administrators create Zones to increase network security, and prevent data loss or corruption, by controlling access between devices or user groups. Zones may be specifically used to create:

a) Barriers between devices that use different operating systems. It is often critical to separate servers and storage devices (see FC-FS-4) with different operating systems because accidental transfer of information from one to another may delete or corrupt data;

b) Logical subsets of closed user groups. Administrators may authorize access rights to specific Zones for specific user groups, thereby protecting confidential data from unauthorized access;

c) Groups of devices that are separate from devices in the rest of a Fabric. Zones allow certain processes to be performed on devices in a group without interrupting devices in other groups; or

d) Temporary access between devices for specific purposes. Administrators may remove Zone restrictions temporarily, then restore Zone restrictions to perform normal processes.

Zones may be configured via this Server, or via a mechanism outside the scope of this standard.

An Nx_Port may be a member of one or more Zones. Zone membership may be specified by:

a) The N_Port_Name of the Nx_Port connected to the Switch;

b) The N_Port_ID assigned during Fabric Login;

c) The Node_Name associated with the Nx_Port; note that the Node_Name may include more than one Nx_Port;

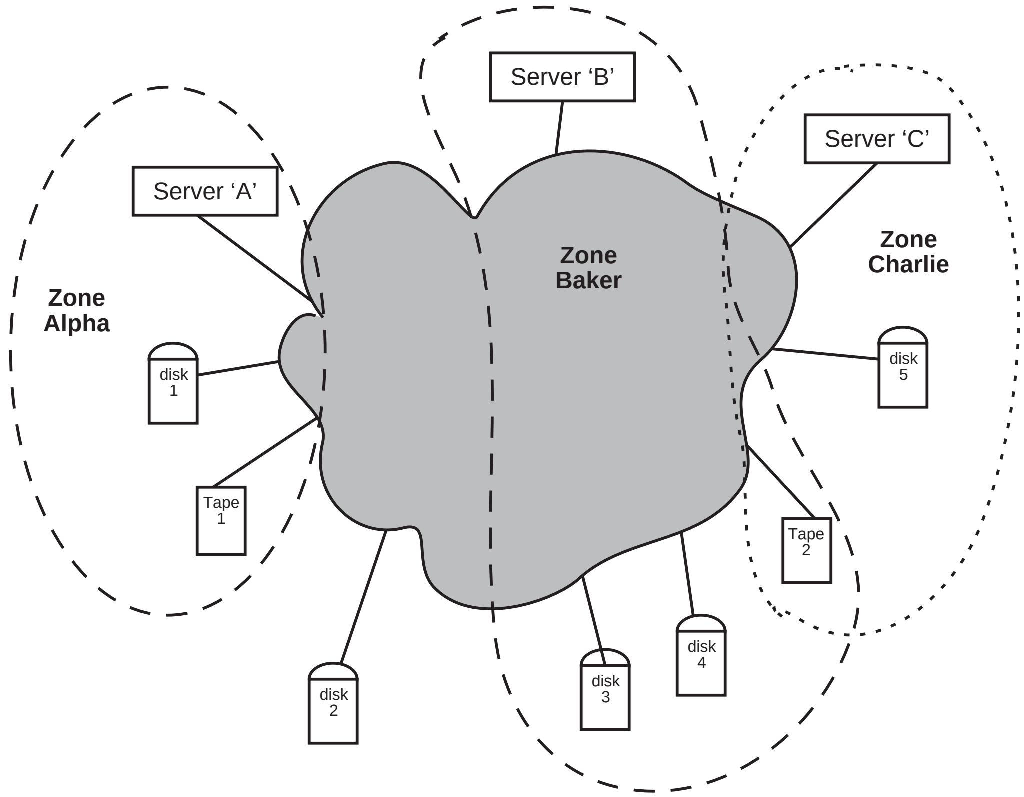
d) The F_Port_Name of the Fx_Port to which the Nx_Port is connected; or

e) The Domain_ID and physical port number of the Switch Port to which the Nx_Port is attached (e.g., as “Domain 1, Port 1”). The format and interpretation of the physical port number is not defined by this standard.

NOTE 32 – The use of Domain_ID and physical port number may cause interoperability issues. The GID_DP Name Server command introduced in FC-GS-4 allows discovery of the N_Port_IDs associated with a Domain_ID and physical port number (see 5.2.5.28).

The Fabric Zone Server may be used to create a Zone by specifying the Zone Members. One or more Zones may be collected into a Zone Set, and a Zone may be a member of more than one Zone Set. A Zone Set creates a collection of Zones that may be activated or deactivated as a single entity across all Switches in the Fabric (e.g., you could have two Zone Sets, one for normal operation, and another for backup during off-hours). Only one Zone Set may be activated at one time.

An example of an Active Zone Set used to restrict visibility amongst different operating system images is shown in figure 14.



	Zone Set		
Zone Set Name	**"OS_partition"**		
Zone Names	**"Alpha"**	**"Baker"**	**"Charlie"**
Zone Members	Server A	Server B	Server C
	disk 1	disk 3	disk 5
	tape 1	tape 2	tape 2
		disk 4	

Figure 14 – Active Zone Set example

Switches are linked through Inter-Switch Links (ISLs) to form multi-Switch Fabrics. In a multi-Switch Fabric, the Active Zone Set applies to the entire Fabric.

See FC-SW-6 to understand how the Zoning information are distributed among the Switches of a multi-Switch Fabric.

6.4.2 Terminology

The terminology used in this document is the following:

a) Active Zone Set: the Zone Set currently enforced by the Fabric;

b) Zone Set Database: the database of the Zone Sets available to be activated within a Fabric; and

c) Zoning Database: a generic term used to indicate both the above concepts.

The relationship between the Zone Set Database and the active Zone Set is shown in figure 15.

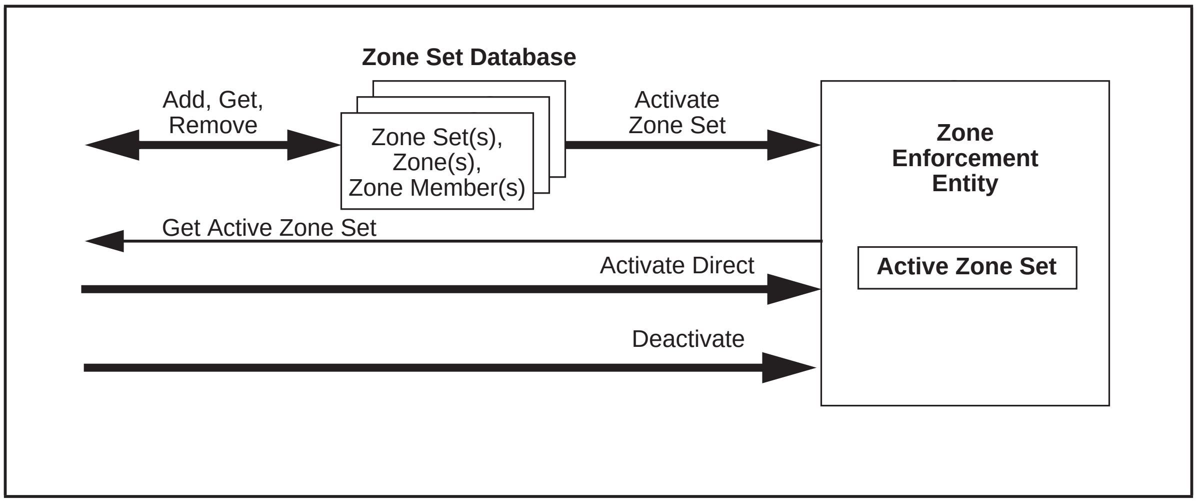


Figure 15 – Zone Set Database and Zone enforcement

To understand how Activate and Deactivate work, it is useful to understand the relationship between the Zone Set Database and the Zone Enforcement Entity. The Zone Set Database contains all of the Zone Sets, Zones, and Zone Members added to it and removed from it.

The Zone Enforcement Entity is an abstraction of the resources within the Fabric that actually take the information and enforce it in a software and/or hardware sense, as appropriate for the implementation.

If no Zone Set has been activated, or if a Zone Set is explicitly deactivated, then there is no enforcement. All Nx_Ports may communicate without restriction, or no Nx_Ports may communicate, depending on the implemented Default Zone behavior.

A Zone Set may be activated using one of two methods. The first method is to use the Add and Remove requests to create a Zone Set in the database, then activate the defined Zone Set.

The second method communicates a Zone Set directly with the Zone Enforcement Entity.

NOTE 33 – The Zone Set is not stored in the database, and therefore Zone and Zone Member objects are not added to or removed from this Zone Set. The contents of this Zone Set may be queried.

In either case, activating a Zone Set implicitly deactivates a previously activated Zone Set.

The Zone Set name of the activated Zone set has no relation to any Zone Set name in the database (i.e., changes to the Zone Set database do not affect the Active Zone Set).

The Zone Enforcement Entity shall retain the Active Zone Set through power cycles and normal initialization processes.

A change in the Active Zone Set shall cause an RSCN to be originated. The RSCN may be sent to all Nx_Ports that are registered to receive RSCN or only to affected Nx_Ports that are registered to receive RSCN. The determination of affected Nx_Ports is outside the scope of this standard.

6.4.3 Protocol

Fabric Zone Server additions, removals, activations, and queries are managed through protocols containing a set of Request CT_IUs and Response CT_IUs supported by the Fabric Zone Server.

For a Fabric Zone Server request, the payload shall be transported from the requestor to the Fabric Zone Server using a Request CT_IU. The corresponding Fabric Zone Server response is transported from the Fabric Zone Server to the requestor, in the Exchange established by the requestor, using a Response CT_IU.

The GS_Subtype shall be set in the CT_IU preamble for Fabric Zone Server request and their responses as indicated in table 125; the Command Code shall be set as specified in 6.4.6.3 and 6.4.7.7.

6.4.4 Zoning Management Framework

From a management perspective, two distinct sets of management requests, Enhanced and Basic, are defined to interact with the Fabric Zone Server. If all the Switches in a Fabric support the Enhanced request set, then it may be used by a management application, otherwise only the Basic request set may be used, in order to support backward compatibility.

Two control management requests are defined to deal with these two distinct request sets:

a) GFEZ (Get Fabric Enhanced Zoning Support) to provide a management application with the ability to query the Fabric about its support for Enhanced Zoning; and

b) SFEZ (Set Fabric Enhanced Zoning Support) to provide a management application with the ability to change the Zoning operational mode of the Fabric from Basic to Enhanced. (See table 272).

Table 272 – Control Zoning Management Requests

Code	Mnemonic	Description	Reference subclause
0142h	GFEZ	Get Fabric Enhanced Zoning Support	6.4.10.1.1
0242h	SFEZ	Set Fabric Enhanced Zoning Support[a]	6.4.10.1.2
[a]Rejected with reason code ‘Unable to perform command request’ with reason code explanation ‘Session established by another Client’ when a Server Session is already in place.			

Before issuing Zoning requests to the Fabric Zone Server, a management application should query the Fabric Zone Server about the Enhanced Zoning capabilities and the operational mode of the Fabric by issuing a GFEZ request.

A Fabric may work in Enhanced Zoning Mode or in Basic Zoning mode.

If the Fabric is working in Enhanced Zoning mode, then the management application shall use the Enhanced Zoning rules, commands and payloads to manage the Zoning Database. Basic Zoning commands that modify the Zoning configuration shall be rejected by the Fabric Zone Server. Basic Zoning queries may be rejected by the Fabric Zone Server.

If the Fabric is working in Basic Zoning mode, then the management application shall use the Basic Zoning rules, commands and payloads to manage the Zoning Database. Enhanced Zoning commands and payload shall be rejected by the Fabric Zone Server.

The model for this behavior is depicted in figure 16.

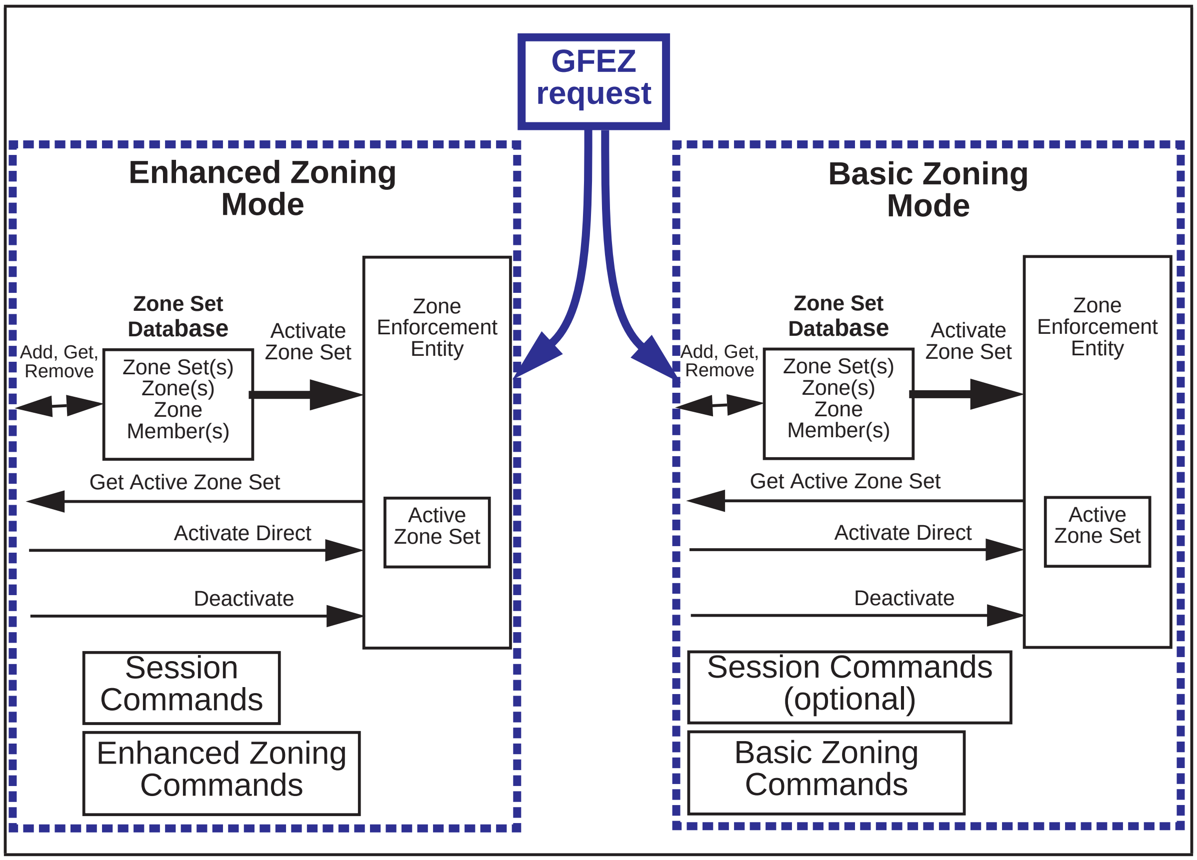


Figure 16 – Zoning Management model

If the Fabric is working in Basic Zoning mode and is able to work in Enhanced mode, then the management application may change the Zoning operational mode of the Fabric by issuing a SFEZ request. If the Fabric is unable to support Enhanced Zoning mode, the management application may discover which Switches in the Fabric do not support Enhanced Zoning through the GFEZ command.

6.4.5 Default Zoning

For both Basic and Enhanced Zoning, there shall exist a Default Zone. The Default Zone has the following properties:

a) Any Nx_Port that is not a member of any Zone in the Active Zone Set shall be a member of the Default Zone;

b) The Fabric may not allow members of the Default Zone to interact with any other members in the Default Zone;

c) The Fabric shall not allow the members of the Default Zone to interact with members of any other Zone; and

d) No mechanisms or methods shall explicitly control Default Zone resources (e.g., members).

NOTE 34 – It is recommended that implementations not provide mechanisms to explicitly monitor Default Zone resources (e.g., members).

6.4.6 Basic Zoning Management

6.4.6.1 Overview

In the context of Basic Zoning Management an interaction with the Fabric Zone Server may occur at any time, without any Server Session. The Fabric does not guarantee the consistency of any returned information, and multiple management applications may manage, at the same time, the Zone Set Database, with unpredictable results. However, if only the Active Zone Set is used the information is consistent.

A management application may delineate a set of Basic Zoning management requests with the SSB and SSE commands. In contrast to Enhanced Zoning (see 6.4.7), Basic Zoning defines no Fabric behavior associated with SSB and SSE. In Basic Zoning, SSB and SSE act only within a Switch (i.e., not Fabric wide). SSB and SSE allow a set of management requests to be related to each other within a single Switch. This relationship may be used to achieve a consistent behavior when two or more management applications operate on a single Switch.

Any actions taken by the Zone Server on its data in response to Basic Zoning commands shall persist through Logout independently of whether the commands are requested within Server Sessions or not, and independently of whether the Server Session is terminated by SSE command or Logout; however, Basic Zoning commands that request changes to Zoning Definitions may be rejected and may cause no action by the Zone Server on its data if another Client has a Server Session to the same Switch.

6.4.6.2 Add and Remove

Fabric Zone Server “add” requests are used to define Zones and Zone Sets within the Fabric’s Zone Set Database. An entire Zone Set and its contained Zones and Zone Members may be defined in a single request; or, Zones and Zone Members may be added to and removed from a defined Zone Set.

The Fabric Zone Server may reject add requests due to Fabric Zone Server resource limitations. However, the Fabric Zone Server shall add all attributes for a given object if it adds any attribute.

The Fabric Zone Server may reject any add or remove requests for reasons not specified in this standard.

A request to remove a Zone Set, Zone, or Zone Member that does not exist shall not be considered an error.

A request to add a Zone to a Zone Set that does not exist, or a request to add a Zone Member to a Zone that does not exist, shall be rejected. A request to add a Zone Set with zero Zones shall not be considered an error.

For Basic Zoning, if all Zones are removed from a Zone Set, that Zone Set shall also be removed. For Basic Zoning, if all Zone Members are removed from a Zone, that Zone shall also be removed. This deletion shall be performed implicitly by the Fabric Zone Server.

A request to add a Zone Member that is not currently attached to the Fabric shall not be considered an error.

If overlapping add and/or remove requests for the same attribute are performed, then the Fabric Zone Server shall, when all requests have completed, leave the attribute as one of the requested attribute values. However, it is indeterminate which of the overlapping requests take precedence.

6.4.6.3 Basic Zoning Management Commands

See table 273 for the CT requests defined to manage the Zoning Database in Basic Zoning Management context.

Table 273 – Fabric Zone Server - Basic Zoning Request Command Codes

Code	Mnemonic	Description	Reference subclause
0100h	GZC	Get Capabilities	6.4.10.3.1
0111h	GEST	Get Enforcement State	6.4.10.3.2
0112h	GZSN	Get Zone Set List	6.4.10.3.3
0113h	GZD	Get Zone List	6.4.10.3.4
0114h	GZM	Get Zone Member List	6.4.10.3.5
0115h	GAZS	Get Active Zone Set	6.4.10.3.6
0116h	GZS	Get Zone Set	6.4.10.3.7
0117h	GAR	Get Activation Results	6.4.10.3.8
0200h	ADZS	Add Zone Set	6.4.10.3.9
0201h	AZSD	Activate Zone Set Direct	6.4.10.3.10
0202h	AZS	Activate Zone Set	6.4.10.3.11
0203h	DZS	Deactivate Zone Set	6.4.10.3.12

Table 273 – Fabric Zone Server - Basic Zoning Request Command Codes (Continued)

Code	Mnemonic	Description	Reference subclause
0204h	AZM	Add Zone Members	6.4.10.3.13
0205h	AZD	Add Zone	6.4.10.3.14
0300h	RZM	Remove Zone Members	6.4.10.3.15
0301h	RZD	Remove Zone	6.4.10.3.16
0302h	RZS	Remove Zone Set	6.4.10.3.17

6.4.7 Enhanced Zoning Management

6.4.7.1 Overview

With Enhanced Zoning Management, a management action (i.e., write access to the Zoning Database) to the Zone Server shall occur only inside a Server Session. This rule applies to both the Zone Set Database and the Active Zone Set. To ensure that Zoning Database information is reported consistently, Query requests to the Fabric Zone Server should be issued inside a Server Session.

The Server Session in Enhanced Zoning is associated with a lock on the Fabric (seeFC-SW-6). The SSB request locks the Fabric, and the SSE request releases the lock.

Enhanced Zoning commands that modify Zoning definitions shall be rejected if they are not requested within a Server Session. Any actions taken by the Zone Server on its data in response to Enhanced Zoning commands shall persist through Logout only if the commands are followed by a CMIT command within the same Server Session, independently of whether the Server Session is terminated by SSE command or Logout.

6.4.7.2 Zone Types

6.4.7.2.1 Hard Zoning

Hard Zoning is managed by setting the Hard Zoning Attribute. When this attribute is not specified, the Zoning enforcement is also not specified, and an implementation may enforce a Zoning configuration in a best effort manner. An implementation may enforce the Zoning configuration on a frame-by-frame basis. Otherwise the implementation shall enforce the Zoning configuration through name server visibility (i.e., Soft Zoning).

When this attribute is specified, the Zone configuration shall be enforced on a frame-by-frame basis. If an implementation is not capable of enforcing the Zone configuration the activation of the Zone configuration shall fail. In case of an activation failure, the CT reject shall return the reason code “Unable to perform command requested”, with a reason code explanation of “Hard Enforcement Failed”.

To allow the coexistence across the same Fabric of Hard and best effort Zones, if Zoning for an Nx_Port is enforced using Hard Zoning enforcement for any Zone, Zoning for that Nx_Port shall be enforced using Hard Zoning enforcement for all zones in which it is a member.

The activation of a Hard Zone (see FC-SW-6) may succeed because some of the Nx_Ports to be zoned are not connected to the Fabric at activation time. When an Nx_Port connects and the Fabric does not have enough resources to continue to enforce the Hard Zone definition, then the Fabric shall reject or discard, as appropriate for the class of service, any frames not addressed to Fabric Services. The addition of a Soft Zone (see FC-SW-6) may also cause an activation to fail, if an Fx_Port is not able to enforce the additional Zone definitions.

This allows Zoning configuration to be defined in where the enforcement is soft for some devices and hard for others. This case may be managed by defining two overlapping Zones, a best effort Zone and an Hard Zone. The best effort Zone defines which Nx_Ports are allowed to communicate. The Hard Zone defines the subset of these Nx_Ports for which Zoning shall be enforced in an hard manner.

Fabrics may be composed of Switches that support hard Zoning and Switches that support only soft Zoning. The Fabric administrator decides which devices have to be managed with hard Zoning, and which ones with soft Zoning.

6.4.7.2.2 Broadcast Zoning

6.4.7.2.2.1 Overview

Broadcast Zoning is enabled by using a broadcast attribute on zones that contain Nx_Ports that send or receive broadcast frames. Use of the broadcast attribute allows devices to send broadcast frames to some devices but not to others.

6.4.7.2.2.2 Zoning Active (i.e., Active Zoneset)

Broadcast frames are delivered to all logged in Nx_Ports that share a Broadcast Zone with the source of the frame (as indicated by the S_ID of the frame). If Zoning is active and no zones have the broadcast attribute, no broadcast frames are delivered. Broadcast Zoning shall be enforced at the destination Switch with respect to each destination Nx_Port.

6.4.7.2.2.3 Zoning Not Active (i.e., no Active Zoneset)

Broadcast frames are delivered to all logged in Nx_Ports.

6.4.7.2.2.4 Broadcast on FL_Ports

If any NL_Port attached to an FL_Port shares a Broadcast Zone with the source of the broadcast frame, or Zoning is not active, the frame shall be sent to ~~the~~ all the devices on the loop (see FC-AL-2).

6.4.7.2.3 Wildcard Zoning

Wildcard Zoning is managed by using the Wildcard Zone Member. The function of the Wildcard Zone Member is to match a subset of the Nx_Ports of a Fabric in a single Zone without the need to list explicitly all members. The Subtype field of the Wildcard Zone Member allows to specify a specific subset of Nx_Ports based on Nx_Port properties from the Name Server. The defined subsets are:

a) Simplified Behavior Nx_Ports: the Wildcard Zone Member matches all Nx_Ports that register the Simplified Behavior declared feature (see Annex A) for FC-4 type DEh in the FC Name Server (see Annex A). If one Zone contains a Wildcard Zone Member matching Simplified Behavior Nx_Ports, then:

    A) that Zone shall contain no other members; and

B) no other Zone in the same Zone Set shall contain a Wildcard Zone Member matching Simplified Behavior Nx_Ports.

6.4.7.2.4 Peer Zoning

Peer Zoning is managed through the definition of one or more Peer Zones. A Peer Zone is a Zone with the Peer Zone Attribute (see 6.4.8.3.8). A Peer Zone identifies a Principal member through the Peer Zone Attribute and a list of Peer members as Zone members (see 6.4.8.3.6). The semantic of a Peer Zone is that:

a) Peer members are allowed to communicate with the Principal member; and

b) Peer members are not allowed to communicate among themselves (unless allowed by other Zones in the Zone Set).

Being Zones, Peer Zones may be defined in both the Active Zone Set and in the Zone Set Database and may be managed as specified in 6.4.7.3. In addition, in order to facilitate the deployment of Peer Zoning in environments where automated storage provisioning is performed, a set of three commands to read, add/update, and remove a Peer Zone into the Active Zone Set is defined:

a) GAPZ: Get Active Peer Zone (see 6.4.10.4.30);

b) AAPZ: Add/Replace Active Peer Zone (see 6.4.10.4.31); and

c) RAPZ: Remove Active Peer Zone (see 6.4.10.4.32).

Being Zones, Peer Zones are identified by a Zone Name. However, specifying a Zone Name may be impractical in environments where automated storage provisioning is performed. In these environments a default Peer Zone Name may be algorithmically computed from the N_Port_Name of the Principal member as ‘X0_YYYYYYYYYYYYYYYY’, where YYYYYYYYYYYYYYYY is the ASCII encoding of the hexadecimal representation of the Principal N_Port_Name.

6.4.7.3 Processing of Zoning Database Changes

With respect to a given Client, the Zoning Database update process begins when the Server Session Begin (SSB) command is accepted by the Fabric Zone server and concludes when the Server Session End (SSE) command is accepted by the Fabric Zone server. During this time the Client is said to be in Update mode.

When the SSB request is received, the Fabric Zone server creates and maintains a temporary instance of the Zoning Database on behalf of the requesting Client. Any changes requested by the Client are then made to the Client’s temporary instance of the Zoning Database. Once the Client has completed all changes and updates to its temporary instance of the database, the Client issues a Commit Zone Changes request to the Fabric Zone server. As part of the commit function, the Fabric Zone server ensures that the changes made to the Client’s temporary instance of the database are propagated through the Fabric. After the SSE request is received, the temporary instance of the Zoning Database is deleted.

As part of the commit processing and prior to any changes being made to the Fabric database, a consistency check is performed on the Zoning database to ensure that the changes are valid. If the changes are valid then the Fabric database is updated accordingly. If the changes are not valid, then the Commit Zone Changes command is failed with the appropriate reason code and reason code explanation.

Examples of not valid changes are any kind of unresolved references present in the Zone Set Database, such as:

a) A Zone Set Object in the Zone Set Database referencing ~~none existent~~ Zone Objects;

b) A Zone Object in the Zone Set Database referencing none existent Zone Alias Objects; or

c) A Zone Object in the Zone Set Database referencing none existent Zone Attribute Objects.

6.4.7.4 Zoning Database Queries and Update Mode

Access to the Fabric Zone server is serialized by the Fabric in that only one Client may be in Update mode at a given time. However, multiple clients may query the Zoning database at a given time. When a Client that is not in Update mode issues queries to the Fabric Zone server, those queries are directed against the Zoning database. When a Client that is in Update mode issues queries to the Fabric Zone server, those queries are directed against the Client's instance of the Zoning database.

6.4.7.5 Management Rules

Management of the Zone Set Database is subject to the following rules:

a) Checks for consistency shall be performed at commit time. If the references in the various Zone Sets and their related objects are not reconciled at commit time then the Commit command request is failed;

b) Incremental updates to the database are not considered complete until the Commit is requested by the Client and completed successfully by the Fabric Zone Server; and

c) Access to the management function is serialized by the Fabric using the session protocol described in FC-SW-6. If a management application does not interact with the Fabric Zone Server for two minutes, then the session shall be automatically released.

Query commands return information based on the Client's view of the Zoning Database. If the Fabric Zone Server is currently maintaining a local instance of the database on behalf of the Client, then the results of the query commands are based on information maintained in the local instance. If no local instance exists, then the results of the query commands are based on the Zoning Database itself. Multiple clients may query the Zoning Database at the same time.

6.4.7.6 Session Commands

See table 274 for the CT requests defined to manage the Server Session in Enhanced Zoning Management context

Table 274 – Fabric Zone Server – Session Request Command Codes

Code	Mnemonic	Description	Reference subclause
7FF9h	SSB	Server Session Begin[a,d]	4.9.5.2
7FFAh	SSE	Server Session End[b,e]	4.9.5.3
7FFBh	CMIT	Commit Zone Changes[c,e]	6.4.10.2.3
7FFCh	SPCMIT	FC-SP Commit Zone Changes[c,e]	see FC-SP-2

[a] Coupled with the ACA SW_ILS processing (see FC-SW-6).
[b] Coupled with the RCA SW_ILS processing (see FC-SW-6).
[c] Coupled with the SFC and UFC SW_ILSs processing (see FC-SW-6).
[d] Rejected with reason code ‘Session could not be established’ when the Fabric Zone Server has a Server Session established by other clients.
[e] Rejected with reason code ‘Unable to perform command request’ with reason code explanation ‘Session established by another Client’ when a Server Session is already in place.

6.4.7.7 Enhanced Zoning Management Commands

See table 275 for the CT requests defined to manage the Zoning Database in Enhanced Zoning Management context

Table 275 – Fabric Zone Server – Enhanced Zoning Request Command Codes

Code	Mnemonic	Description	Reference subclause
0117h	GAR	Get Activation Results	6.4.10.3.8
0120h	GZA	Get Zone Attribute Object Name	6.4.10.4.2
0121h	GZAB	Get Zone Attribute Block	6.4.10.4.3
0122h	GZSE	Get Zone Set List - Enhanced	6.4.10.4.4
0123h	GZDE	Get Zone List - Enhanced	6.4.10.4.5
0124h	GZME	Get Zone Member List - Enhanced	6.4.10.4.6

[a] The Zone Set defined in the Request CT_IU is not added to the Zone Set Database.
[b] Rejected with reason code ‘Unable to perform command request’ with reason code explanation ‘Session established by another Client’ when a Server Session is already in place.

Table 275 – Fabric Zone Server – Enhanced Zoning Request Command Codes (Continued)

Code	Mnemonic	Description	Reference subclause
0125h	GZAL	Get Zone Attribute Object List	6.4.10.4.7
0126h	GAZSE	Get Active Zone Set - Enhanced	6.4.10.4.26
0128h	GAL	Get Alias List	6.4.10.4.8
0129h	GAM	Get Alias Member List	6.4.10.4.9
012Ah	GAPZ	Get Active Peer Zone	6.4.10.4.30
012Bh	GAN	Get Alias Names	6.4.10.4.30
0211h	AZSDE[b]	Activate Zone Set Direct - Enhanced[a]	6.4.10.4.9
0212h	AZSE[b]	Activate Zone Set - Enhanced	6.4.10.4.27
0213h	DZSE[b]	Deactivate Zone Set - Enhanced	6.4.10.4.29
0220h	CZS[b]	Create Zone Set	6.4.10.4.12
0221h	AZ[b]	Add Zones	6.4.10.3.9
0224h	AZME[b]	Add Zone Members - Enhanced	6.4.10.4.18
0225h	CZ[b]	Create Zone	6.4.10.4.13
0226h	CZA[b]	Create Zone Attribute Object	6.4.10.4.15
0227h	SZA[b]	Set Zone Attribute Object Name	6.4.10.4.10
0228h	SZAB[b]	Set Zone Attribute Block	6.4.10.4.11
0229h	CA[b]	Create Alias	6.4.10.4.14
022Ah	AAM[b]	Add Alias Members	6.4.10.4.20
022Bh	AAPZ	Add/Replace Active Peer Zone	6.4.10.4.31
0321h	RZ[b]	Remove Zones	6.4.10.4.17
0324h	RZME[b]	Remove Zone Members - Enhanced	6.4.10.4.19
0325h	RAPZ	Remove Active Peer Zone	6.4.10.4.32
032Ah	RAM[b]	Remove Alias Members	6.4.10.4.21
032Bh	DLZS[b]	Delete Zone Set	6.4.10.4.22

[a] The Zone Set defined in the Request CT_IU is not added to the Zone Set Database.

[b] Rejected with reason code ‘Unable to perform command request’ with reason code explanation ‘Session established by another Client’ when a Server Session is already in place.

Table 275 – Fabric Zone Server – Enhanced Zoning Request Command Codes (Continued)

Code	Mnemonic	Description	Reference subclause
032Ch	DLZ[b]	Delete Zone	6.4.10.4.23
032Dh	DLA[b]	Delete Alias	6.4.10.4.24
032Eh	DLZA[b]	Delete Zone Attribute Object	6.4.10.4.25

[a] The Zone Set defined in the Request CT_IU is not added to the Zone Set Database.
[b] Rejected with reason code ‘Unable to perform command request’ with reason code explanation ‘Session established by another Client’ when a Server Session is already in place.

6.4.8 Zoning Data Structures

6.4.8.1 General Name Format

Several Zoning objects are identified through a name. Unless otherwise specified, those names shall follow the general name format described in table 276:

Table 276 – General Name Format

Item	Size
Name Length (n)	1
Reserved	3
Name Value	n
Fill Bytes (m)	0,1,2, or,3

6.4.8.1.1 Name Length

This field specifies the length in bytes of the Name Value plus any required fill bytes. This value shall be a multiple of four.

6.4.8.1.2 Name Value

The Name Value field shall contain the ASCII characters that specify the name, not including any required fill bytes. Names shall adhere to the following rules:

a) A name shall be between 1 and 64 characters in length (n);

b) All characters shall be 7 bit ASCII characters;

c) The first character of a given name shall be a letter. A letter is defined as either an upper case (A to Z) character or a lower case (a to z) character; and

d) Any character other than the first character shall be a lower case character (a to z), an upper case character (A to Z), a number (0 to 9), or one of the symbols in table 277.

Table 277 – Name Value Characters

Value	Symbol	Description
24h	**$**	Dollar sign
28h	**(**	Open parenthesis
29h	**)**	Close parenthesis
2Dh	**-**	Hyphen
2Eh	**.**	Period
5Eh	**^**	Circumflex accent
5Fh	_	Underline

6.4.8.1.3 Fill Bytes

In order to ensure that the Name field length is a multiple of four, fill bytes are added as necessary to the end of the Name Value. Fill bytes shall be 00h. The number of fill bytes (m) is zero, one, two, or three depending on the length of the actual name (n). Therefore the total length of the name field is (n+m).

6.4.8.2 Basic Zoning Data Structures

6.4.8.2.1 Overview

In Basic Zoning framework the Fabric Zone Server object model is shown in figure 17.

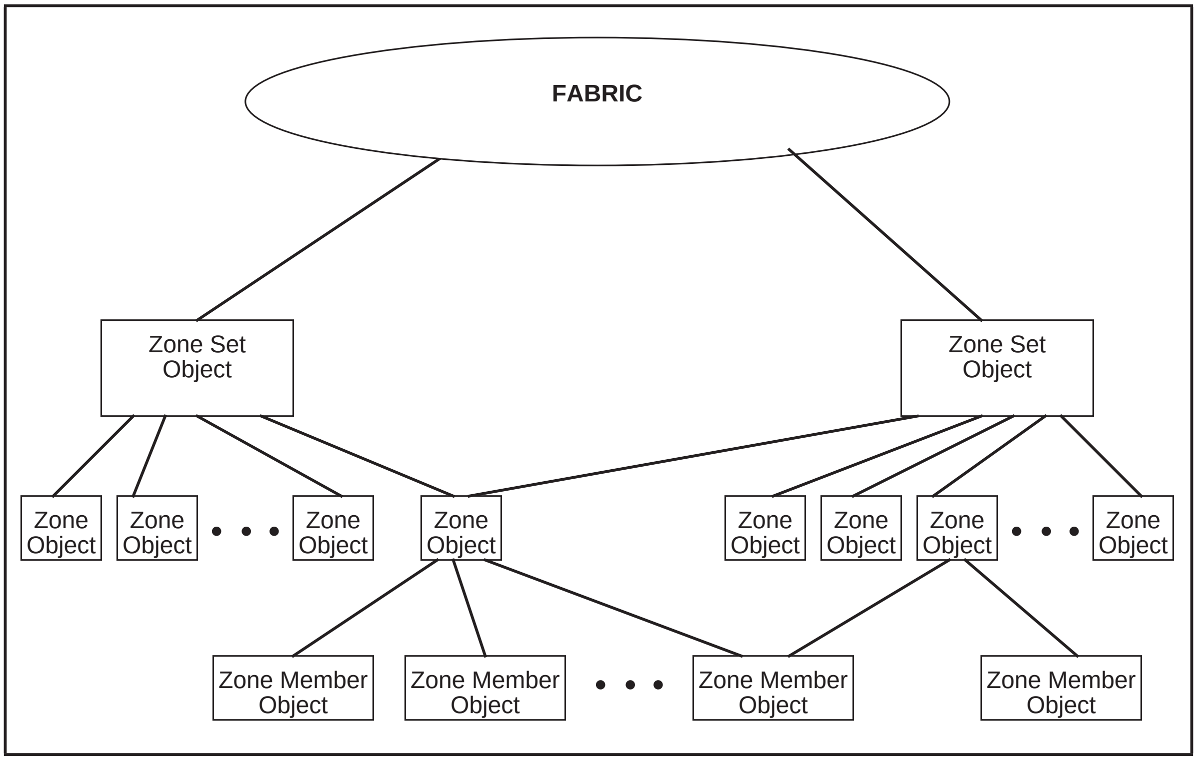


Figure 17 – Fabric Zone Server Object Model

The base object class managed by the Fabric Zone Server is the Zone Set object. One or more Zone Set objects belong to a Fabric. Zone Set objects have one or more associated Zone objects. Zone objects have one or more associated Zone Member objects. A Zone may be a member of more than one Zone Set. A Zone Member may be a member of more than one Zone. Zone Set objects, Zone objects, and Zone Member objects may have attributes associated with them, as shown in figure 18.

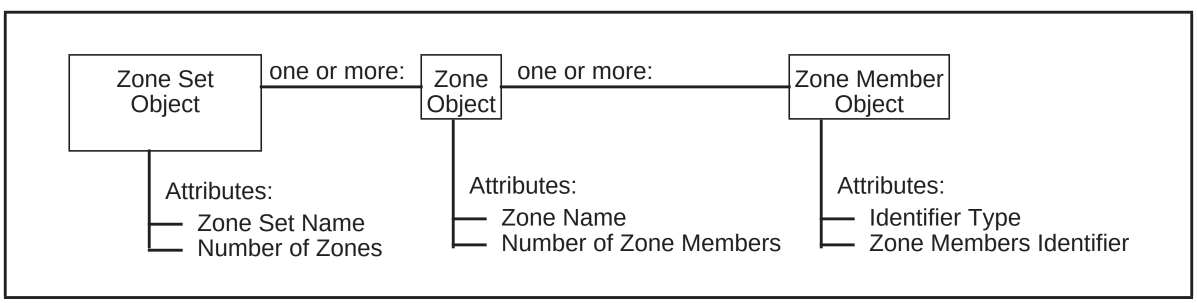


Figure 18 – Zone Set, Zone, and Zone Member attributes

6.4.8.2.2 Zone Set Object

6.4.8.2.2.1 Zone Set Name

The format of the Zone Set Name attribute shall follow the general name format described in 6.4.8.1.

6.4.8.2.2.2 Number of Zones

The format of the Number of Zones attribute shall be as shown in table 278. This attribute shall contain the integer number of zones in the Zone Set.

Table 278 – Number of Zones Format

Item	Size (Bytes)
Number of Zones	4

This attribute may be defined using the protocol described in 6.4.6.

6.4.8.2.3 Zone Object

6.4.8.2.3.1 Zone Name

The format of the Zone Name attribute shall follow the general name format described in 6.4.8.1.

6.4.8.2.3.2 Number of Zone Members

The format of the Number of Zone Members attribute shall be as shown in table 279. This attribute shall contain the integer number of members in the specified Zone.

Table 279 – Number of Zone Members Format

Item	Size (Bytes)
Number of Zone Members	4

This attribute may be defined using the protocol described in 6.4.6. The null value for the Number of Zone Members attribute is 00 00 00 00h.

6.4.8.2.4 Zone Member Object

6.4.8.2.4.1 Zone Member Identifier Type

The values of the Zone Member Identifier Type attribute shall be as shown in table 280. This attribute establishes the format of the information contained in the Zone Member Identifier attribute.

Table 280 – Zone Member Identifier Type- encoding

Encoded value	Description
01h	N_Port_Name
02h	Domain_ID and physical port
03h	N_Port_ID
04h	Node_Name
all others	Reserved

NOTE 35 – The values specified in table 280 differ from those specified in FC-SW-6 for backward compatibility.

NOTE 36 – Fabric rebuilds may result in Domain IDs being reassigned resulting in unpredictable Zoning enforcement behavior for Zone Member Identifier Types 02 and 03. Loop Initialization events may result in arbitrated loop address reassignment resulting in unpredictable Zoning enforcement behavior for Zone Member Identifier Type 03.

If the Fabric Zone Server receives a request containing unsupported member types, that request shall be rejected.

This attribute may be defined using the protocol described in 6.4.6.

NOTE 37 – It is recommended that one value of Zone Member Identifier Type be used for the entire Fabric. Mixing Zone Member Identifier Types may result in overlaps and ambiguities. The resolution of any resulting overlap or ambiguity is not defined by this standard.

NOTE 38 – The use of Type code 02h may result in interoperability issues. The GID_DP Name Server command introduced in FC-GS-4 allows discovery of the N_Port_IDs associated with a Domain_ID and physical port number (see 5.2.5.28).

6.4.8.2.4.2 Zone Member Identifier

The format of the Zone Member Identifier attribute is indicated by the Zone Member Identifier Type. For any Zone Member Identifier Type and format, the specific value of these bytes shall not be restricted by the Fabric Zone Server.

The N_Port_Name Zone Member Identifier format shall be as shown in table 281.

Table 281 – Zone Member Identifier Format - N_Port_Name

Item	Size (Bytes)
N_Port_Name	8

The Domain_ID and Port Zone Member Identifier format shall be as shown in table 282.

Table 282 – Zone Member Identifier Format - Domain_ID and Port

Item	Size (Bytes)
Domain_ID	1
Physical Port number	2
Reserved	5

The N_Port_ID Zone Member Identifier format shall be as shown in table 283.

Table 283 – Zone Member Identifier Format - N_Port_ID

Item	Size (Bytes)
Reserved	1
N_Port_ID	3
Reserved	4

The Node_Name Zone Member Identifier format shall be as shown in table 284.

Table 284 – Zone Member Identifier Format - Node_Name

Item	Size (Bytes)
Node_Name	8

6.4.8.3 Enhanced Zoning Data Structures

6.4.8.3.1 Zone Set Database

The currently defined Zoning Object types are:

a) Zone Set Objects;

b) Zone Objects;

c) Zone Alias Objects; and

d) Zone Attribute Objects.

The Zoning Objects have an Object Identifier as defined in table 285.

Table 285 – Object Identifier Value

Object Identifier	Value
Zone Set Object	01
Zone Object	02
Zone Alias Object	03
Zone Reference Object	04
Zone Attribute Object	05
Reserved	All Others

The Zone Set Database shall not contain the Active Zone Set. The Zone Set Database may contain multiple Zoning objects. Objects defined in the Zone Set Database need not be referenced. In the Zone Set Database the Zoning objects may reference each other using names formatted as specified in 6.4.8.1.

Figure 19 depicts the logical structure of the Zone Set Database.

Each Zone Set definition references its Zone objects. Each Zone may reference a Zone Attribute Object or, in the member definitions, some Zone Alias objects.

6.4.8.3.2 Active Zone Set

References are not allowed in the Active Zone Set. At activation time any reference present in a Zone Set or Zone definition in the Zone Set Database shall be resolved. The resulting logical structure of the Active Zone Set is depicted in figure 20.

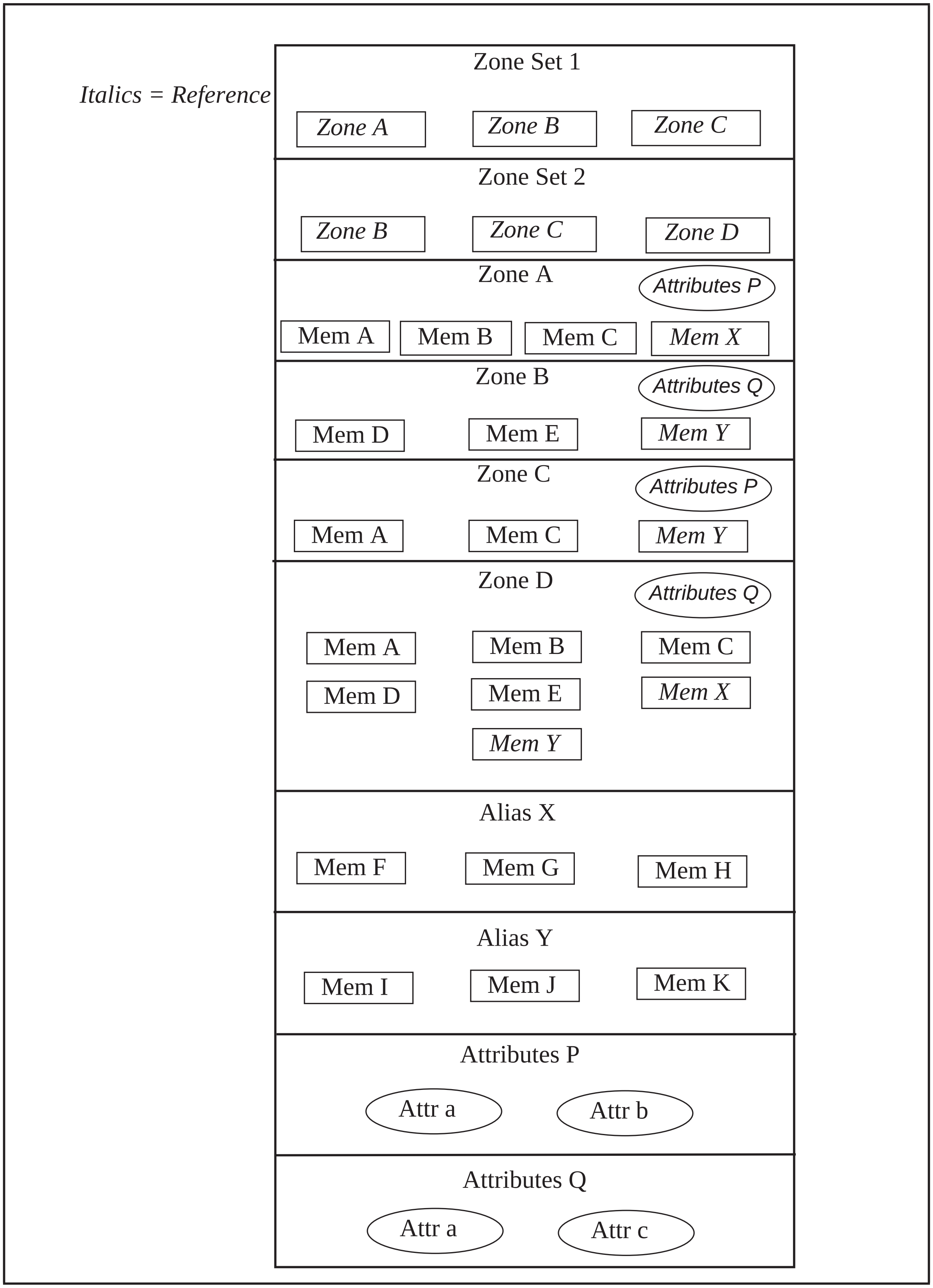


Figure 19 – Logical Structure of the Zone Set Database

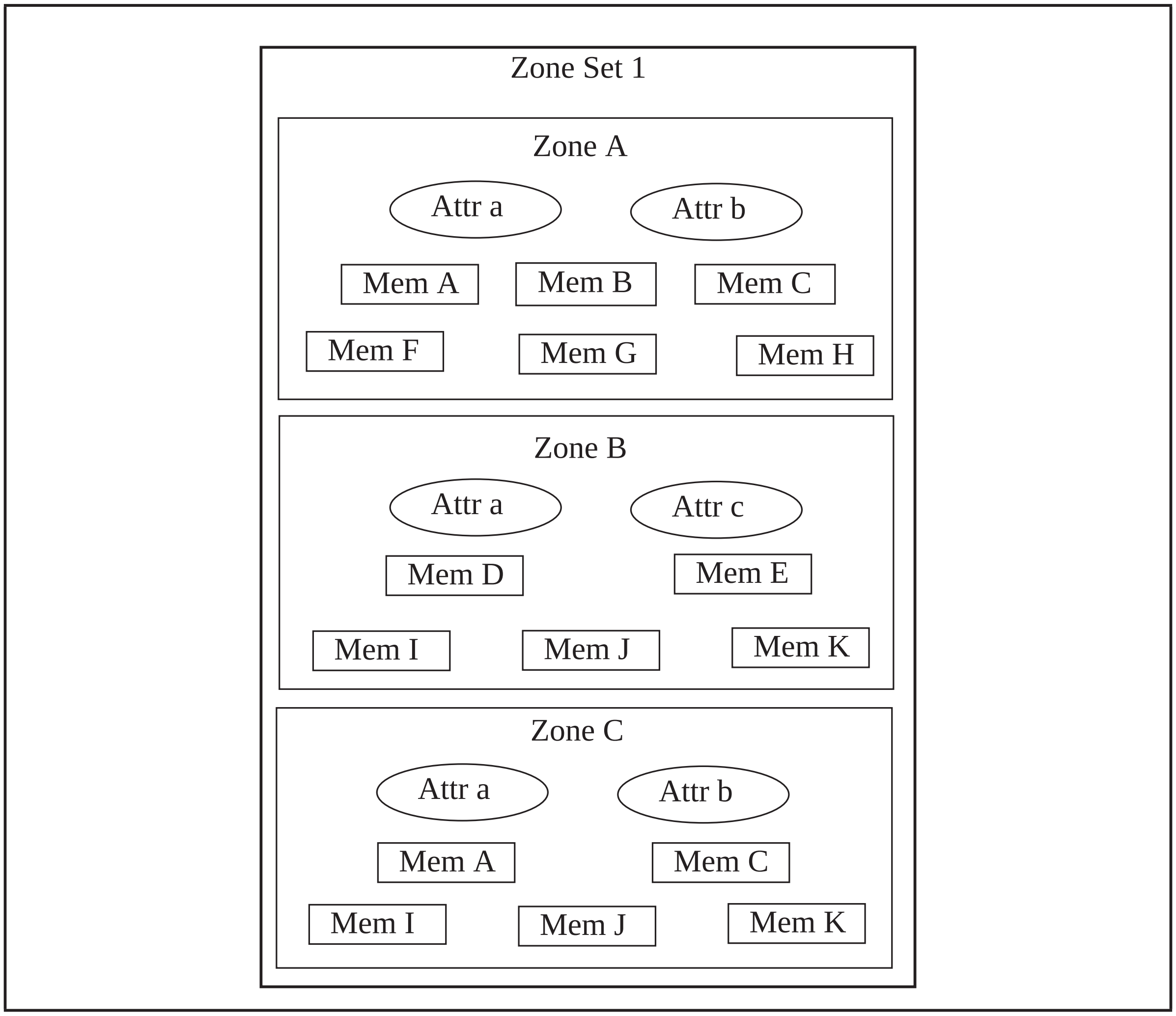


Figure 20 – Logical Structure of the Active Zone Set

6.4.8.3.3 Zone Set Object

6.4.8.3.3.1 Zone Set Object in the Zone Set Database

The Zone Set Object used in the Zone Set Database has the attributes described below and the format depicted in table 286.

Table 286 – Zone Set Object in the Zone Set Database

Item	Value	Size (bytes)
Object Identifier	01	1
Reserved	00	3
Zone Set Name	general name	see 6.4.8.1
Number of Zone References	n	4
Zone Reference #1	Zone Reference	x
Zone Reference #2	Zone Reference	y
...	...	...
Zone Reference #n	Zone Reference	z

Zone Set Name: The format of the Zone Set Name attribute shall follow the general name format described in 6.4.8.1.

Number of Zone References: This attribute shall contain the integer number (expressed over 4 bytes) of Zone references in the Zone Set.

Zone Reference: Only Zone Reference objects (Zone Set Member Type hex’04’) are allowed in the Zone Set Object when used in the Zone Set Database.

6.4.8.3.3.2 Zone Set Object in the Active Zone Set

The Zone Set Object used in the Active Zone Set has the attributes described below and the format depicted in table 287.

Table 287 – Zone Set Object in the Active Zone Set

Item	Value	Size (bytes)
Object Identifier	01	1
Reserved	00	3
Number of Zones	n	4
Zone #1	Zone Object	x
Zone #2	Zone Object	y
...	...	...
Zone #n	Zone Object	z

Number of Zones: This attribute shall contain the integer number (expressed over 4 bytes) of zones in the Zone Set.

Zone: Only Zone objects (Zone Set Member Type hex’02’) are allowed in the Zone Set Object when used in the Active Zone Set.

6.4.8.3.4 Zone Reference Object

The Zone Reference Object format is depicted in table 288.

Table 288 – Zone Reference Object

Item	Value	Size (bytes)
Object Identifier	04	1
Reserved	00	3
Zone Name	general name	see 6.4.8.1

Zone Name: The format of the Zone Name attribute shall follow the general name format described in 6.4.8.1.

6.4.8.3.5 Zone Object

6.4.8.3.5.1 Zone Object in the Zone Set Database

The Zone Object used in the Zone Set Database allows references to other objects. It has the attributes described below and the format depicted in table 289.

Table 289 – Zone Object in the Zone Set Database

Item	Value	Size (bytes)
Object Identifier	02	1
Reserved	00	3
Zone Name	general name	see 6.4.8.1
Zone Attribute Object Name	general name	see 6.4.8.1
Number of Zone Members	n	4
Zone Member #1	Zone Member object	see 6.4.8.3.6
Zone Member #2	Zone Member object	see 6.4.8.3.6
...	...	...
Zone Member #n	Zone Member object	see 6.4.8.3.6

Zone Name: The format of the Zone Name attribute shall follow the general name format described in 6.4.8.1.

Zone Attribute Object Name: The format of the Zone Attribute Object Name attribute shall follow the general name format described in 6.4.8.1. Its value references the Zone Attribute Object whose attributes apply for the Zone. A null value 00 00 00 00 00 00 00 00h indicates that no attributes are associated with that Zone definition.

Number of Zone Members: This attribute shall contain the integer number (expressed over 4 bytes) of Zone Members in the Zone.

Zone Member: See 6.4.8.3.6.

6.4.8.3.5.2 Zone Object in the Active Zone Set

The Zone Object used in the Active Zone Set does not allow references to other objects. It has the attributes described in this subclause and the format depicted in table 290.

Table 290 – Zone Object in the Active Zone Set

Item	Value	Size (bytes)
Object Identifier	02	1
Reserved	00	3
Zone Name	general name	see 6.4.8.1
Number of Zone Attribute Entries	m	4
Zone Attribute Entry #1	Zone Attribute Entry	see table 300
Zone Attribute Entry #2	Zone Attribute Entry	see table 300
...	...	...
Zone Attribute Entry #m	Zone Attribute Entry	see table 300
Number of Zone Members	n	4
Zone Member #1	Zone Member object	see 6.4.8.3.6
Zone Member #2	Zone Member object	see 6.4.8.3.6
...	...	...
Zone Member #n	Zone Member object	see 6.4.8.3.6

Zone Name: The format of the Zone Name attribute shall follow the general name format described in 6.4.8.1.

Number of Zone Attribute Entries: This attribute shall contain the integer number of Zone Attribute Entries in the Zone.

Zone Attribute Entry: See 6.4.8.3.8.

Number of Zone Members: This attribute shall contain the integer number of Zone Members in the Zone.

Zone Member: See 6.4.8.3.6.

6.4.8.3.6 Zone Member Object

The Zone Member Object has the format depicted in table 291 and the attributes described below.

Table 291 – Zone Member Object Format

Item	Size (Bytes)
Zone Member Identifier Type	1
Reserved	3
Zone Member Identifier value	x

Zone Member Identifier Type: The values of the Zone Member Identifier Type attribute shall be as shown in table 292. This attribute establishes the format of the information contained in the Zone Member Identifier attribute.

Table 292 – Zone Member Identifier Type- encoding

Encoded value	Description
01h	N_Port_Name
02h	Domain_ID and physical port
03h	N_Port_ID
04h	Node_Name
05h	Alias Name
06h	F_Port_Name
07h	Wildcard
E0h to FFh	Vendor Specific
all others	Reserved

If the Fabric Zone Server receives a request containing some unsupported member types, that request shall be rejected.

NOTE 39 – It is recommended that one value of Zone Member Identifier Type be used for the entire Fabric. Mixing Zone Member Identifier Types may result in overlaps and ambiguities. The resolution of any resulting overlap or ambiguity is not defined by this standard.

NOTE 40 – The use of Type code 02h may result in interoperability issues. The GID_DP Name Server command introduced in FC-GS-4 allows discovery of the N_Port_IDs associated with a Domain_ID and physical port number (see 5.2.5.28).

When used in the Zone Set Database, any Zone Member Identifier type may be used as Zone Member.

When used in the Active Zone Set, Alias Names are not allowed as Zone Member Identifier types. Any Alias Name present in a Zone definition in the Zone Set Database shall be resolved in the list of its Alias Members at activation time.

Zone Member Identifier: The format of the Zone Member Identifier attribute is indicated by the Zone Member Identifier Type. For any Zone Member Identifier Type and format the specific value of these bytes shall not be restricted by the Fabric Zone Server.

The N_Port_Name Zone Member Identifier format shall be as shown in table 281.

The Domain_ID and Port Zone Member Identifier format shall be as shown in table 282.

The N_Port_ID Zone Member Identifier format shall be as shown in table 283.

The Node_Name Zone Member Identifier format shall be as shown in table 284.

The F_Port_Name Zone Member Identifier format shall be as shown in table 293.

Table 293 – Zone Member Identifier Format - F_Port_Name

Item	Size (Bytes)
F_Port_Name	8

The Alias Name Member Identifier format is a name formatted as described in 6.4.8.1, and referencing a Zone Alias object.

The Wildcard Zone Member Identifier format shall be as shown in table 294.

Table 294 – Zone Member Identifier Format - Wildcard

Item	Size (Bytes)
Subtype	2
Flags	2
Parameter	4

Subtype: defines the subset of Nx_Ports matched by the Wildcard Zone Member. Subtype values are defined in table 295.

Table 295 – Wildcard Zone Member - Subtype Values

Value	Description
0001h	Simplified Behavior Nx_Ports (see 6.4.7.2.3)

Flags: Reserved.

Parameter: Reserved.

The Vendor Specific Zone Member Identifier format is depicted in table 296.

Table 296 – Zone Member Identifier Format - Vendor Specific

Item	Size (Bytes)
Vendor Identifier	8
Vendor Specifc Value Length	4
Vendor Specific Value	n
Pad	m

Vendor Identifier: Contains the T10 Vendor ID (see 3.6) of the vendor that defines the content of the Vendor Specific Value field.

Vendor Specific Value Length: This field contains the length, in bytes, of the Vendor Specific Value field plus the length of the Pad field.

Vendor Specific Value: This field contains the Vendor Specific Value.

Pad: Fill bytes are added as necessary to the end of the Vendor Specific Value in order to ensure that the total length of the Vendor Specific Zone Member is a multiple of four. Fill bytes shall be 00h. The number of fill bytes (m) is zero, one, two, or three depending on the length of the actual value (n).

6.4.8.3.7 Zone Alias Object

The Zone Alias Object has the attributes described below and the format depicted in table 297.

Table 297 – Zone Alias Object

Item	Value	Size (bytes)
Object Identifier	03	1
Reserved	00	3
Zone Alias Name	general name	see 6.4.8.1.2
Number of Zone Alias Members	n	4
Zone Alias Member #1	Zone Member object	see 6.4.8.3.6
Zone Alias Member #2	Zone Member object	see 6.4.8.3.6
...	...	...
Zone Alias Member #n	Zone Member object	see 6.4.8.3.6

Zone Alias Name: The format of the Zone Alias Name attribute shall follow the general name format described in 6.4.8.1.

Number of Zone Alias Members: This attribute shall contain the integer number of Zone Alias Members in the Zone Alias.

Zone Alias Member: The Zone Alias Member has the format described in 6.4.8.3.6. All Zone Member Identifier Types may be used, with the exception of the Alias Name member (i.e. type ‘05’).

6.4.8.3.8 Zone Attribute Object

The Zone Attribute object is a variable length structure that contains extensible attributes that may be associated with some Zones. The format of the Zone Attribute object is depicted in table 298.

Table 298 – Zone Attribute Object

Item	Value	Size (bytes)
Object Identifier	05	1
Reserved	00	3
Zone Attribute Object Name	general name	see 6.4.8.1.2
Zone Attribute Block	Zone Attribute Block (see table 299)	w

Table 299 – Zone Attribute Block

Item	Size (Bytes)
Number of Zone Attribute Entries	4
Zone Attribute Entry #1	x
Zone Attribute Entry #2	y
...	...
Zone Attribute Entry #n	z

Zone Attribute Object Name: The format of the Zone Attribute Object Name attribute shall follow the general name format described in 6.4.8.1.

Number of Zone Attributes Entries: This field specifies the number of Zone Attribute Entries contained in the Zone Attribute Block. A value of zero in this field shall indicate that no attributes are registered.

Zone Attribute Entry: One Zone Attribute Entry shall be returned for each attribute assigned to the specified Zone Attribute Object. The format of the Zone Attribute Entry is depicted in table 300.

Table 300 – Zone Attribute Entry

Item	Size (Bytes)
Zone Attribute Type	2
Zone Attribute Length	2
Zone Attribute Value	x

Zone Attribute Type: This field indicates the attribute entry type. Valid Zone Attribute Types are depicted in table 301 and shall be restricted to a value between 0000h and 00FFh.

Table 301 – Zone Attribute Types

Zone Attribute Type	Description	Zone Attribute Value
0001h	Protocol[a]	Yes
0002h	Hard Zone[a]	None
0003h	Broadcast Zone[a]	None
0004h	IFR Zone[b]	None
0005h	Peer Zone	Yes
00E0h	Vendor Specific	Yes
other values	Reserved	

[a] For a definition of the Protocol, Hard Zone, and Broadcast Zone attribute types, see FC-SW-6.
[b] For a definition of the IFR Zone attribute type, see FC-IFR.

Zone Attribute Length: This field indicates the total length of the Zone Attribute Entry. This length shall be a multiple of four and includes the following fields:

a) Zone Attribute Type;

b) Zone Attribute Length; and

c) Zone Attribute Value.

Zone Attribute Value: This field specifies the actual attribute value if one exists for the specified Zone Attribute. If present, Attribute Values shall be at least four bytes in length and the length shall be a multiple of four. For Attribute Value fields, fill bytes are added as necessary to the end of the actual value in order to ensure that the length of the value field is a multiple of four. Fill bytes shall be 00h. The number of fill bytes (m) is zero, one, two, or three depending on the length of the actual value (n). Therefore the total length of the value field is (n+m). The format of this field is determined by the Zone Attribute Type field (see table 301).

When a Protocol Attribute Type is specified, the format of the Protocol Attribute Value is depicted in table 302.

Table 302 – Protocol Attribute Value

Item	Size (Bytes)
FC-4 Type	1
Reserved	3

FC-4 Type: The FC-4 type (see FC-FS-4) for which protocol Zoning is enforced. Valid values are 01h to FFh. Device_Data and FC-4 Link_Data frames not having the specified FC-4 Type value shall not be transmitted between members of the Zone. All other frames shall be transmitted between members of the Zone.

When a Peer Zone Attribute type is specified, the format of the Peer Zone Attribute Value is depicted in table 303.

Table 303 – Peer Zone Attribute Value

Item	Size (Bytes)
Principal N_Port_Name	8

Principal N_Port_Name: The N_Port_Name of the Principal member of a Peer Zone (see 6.4.7.2.4).

When a Vendor Specific Attribute Type is specified, the format of the Vendor Specific Attribute Value is depicted in table 304.

Table 304 – Vendor Specific Attribute Value

Item	Size (Bytes)
Vendor Identifier	8
Vendor Specific Value	n
Pad	m

Vendor Identifier: Contains the T10 Vendor ID of the vendor that defines the content of the Vendor Specific Value field.

Vendor Specific Value: This field contains the Vendor Specific Value.

Pad: Fill bytes are added as necessary to the end of the Vendor Specific Value in order to ensure that the total length of the Vendor Specific Zone Member is a multiple of four. Fill bytes shall be 00h. The number of fill bytes (m) is zero, one, two, or three depending on the length of the actual value (n).

6.4.9 Reason code explanations

A Reject CT_IU (see 4.4.4) shall notify the requestor that the request has been unsuccessfully completed. The first error condition encountered shall be the error reported by the Reject CT_IU.

If a valid Fabric Zone Server request is not received, the request is rejected with a reason code of “Invalid command code” and a reason code explanation of “No additional explanation”.

If a Fabric Zone Server request is rejected with a reason code of ‘Unable to perform command request’, then one of the reason code explanations, shown in table 305, is returned.

Table 305 – Reject CT_IU Reason code explanations

Encoded value	Description
00h	No additional explanation
01h	Zones not supported
10h	Zone Set Name unknown
11h	No Zone Set active
12h	Zone Name unknown
13h	Zone State unknown
14h	Incorrect payload length
15h	Zone Set to be activated too large
16h	Deactivate Zone Set failed
17h	Request not supported
18h	Capability not supported
19h	Zone Member Identifier Type not supported
1Ah	Invalid Zone Set definition
20h	Enhanced Zoning Commands Not Supported
21h	Zone Set Already Exists
22h	Zone Already Exists
23h	Alias Already Exists
24h	Zone Set Unknown
25h	Zone Unknown
26h	Alias Unknown
28h	Unknown Zone Attribute Type

Table 305 – Reject CT_IU Reason code explanations (Continued)

Encoded value	Description
29h	Fabric Unable to Work in Enhanced Mode
30h	Basic Zoning Commands Not Supported
31h	Zone Attribute Object Already Exists
32h	Zone Attribute Object Unknown
33h	Request in Process
34h	CMIT in Process
35h	Hard Enforcement Failed
36h	Unresolved references in the Zone Set Database
37h	Consistency checks failed
38h	Provided Zone is Not a Peer Zone
39h	Existing Zone is Not a Peer Zone
Others	Reserved

If a Fabric Zone Server Request is not supported, management applications should be prepared to receive either:

a) Reject reason code 09h, Unable to perform command request, reason code explanation 17h - Request not supported; or

b) Reject reason code 01h - Invalid command code, reason code explanation 18h - Capability not supported.

6.4.10 Zoning Management Commands

6.4.10.1 Control Zoning Commands

6.4.10.1.1 Get Fabric Enhanced Zoning Support (GFEZ)

The Fabric Zone Server shall, if it receives a GFEZ request, return the Enhanced Zoning capabilities supported by the Fabric. The format of the GFEZ Request CT_IU is shown in table 306.

Table 306 – GFEZ Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

The format of the Accept CT_IU to a GFEZ request is shown in table 307.

Table 307 – Accept CT_IU to GFEZ Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Fabric Enhanced Zoning support flags	4
Reserved	3
Number of Switches in the Fabric	1
Switch #1 Enhanced Zoning support entry	12
Switch #2 Enhanced Zoning support entry	12
...	
Switch #n Enhanced Zoning support entry	12

The format of the Switch Enhanced Zoning support entry is shown in table 308.

Table 308 – Switch Enhanced Zoning support entry format

Item	Size (Bytes)
Switch_Name	8
Switch Enhanced Zoning support flags	4

The Fabric Enhanced Zoning support flags are shown in table 309.

Table 309 – Fabric Enhanced Zoning support flags

Bit Position	Description
0	Enhanced Zoning supported. When this bit is set to one, the Fabric is able to support Enhanced Zoning mode. When this bit is set to zero, the Fabric is not able to support Enhanced Zoning mode.
1	Enhanced Zoning enabled. When this bit is set to one, the Fabric is working in Enhanced Zoning mode. When this bit is set to zero, the Fabric is working in Basic Zoning mode.
2	Merge Control Setting. When this bit is set to one the Fabric is working in Restrict mode, so a Switch may join the Fabric only if its Zoning Database is equal to the Fabric’s Zoning Database. When this bit is set to zero the Fabric is working in Allow mode, so a Switch may join the Fabric only if its Zoning Database is able to merge with the Fabric’s Zoning Database.
3	Default Zone Setting. When this bit is set to one the Fabric denies traffic between members of the Default Zone. When this bit is set to zero the Fabric permits traffic between members of the Default Zone.
4	Zone Set Database supported. When this bit is set to one, the Fabric Zone Server is able to maintain a Zone Set Database. When this bit is set to zero, the Fabric Zone Server is not able to maintain a Zone Set Database.
5	Zone Set Database enabled. When this bit is set to one, the Fabric Zone Server is maintaining a Zone Set Database. When this bit is set to zero, the Fabric Zone Server is not maintaining a Zone Set Database.
6	Activate Direct command supported. When this bit is set to one, all the Switches of the Fabric support the Activate Direct command. When this bit is set to zero, at least one Switch of the Fabric does not support the Activate Direct command.
7	Hard Zoning command supported. When this bit is set to one, all the Switches of the Fabric support Hard Zoning. When this bit is set to zero, at least one Switch of the Fabric does not support Hard Zoning.
8	FC-SP Zoning Supported (see FC-SP-2).
9	FC-SP Zoning Enabled (see FC-SP-2).
10 to 31	Reserved

The Switch Enhanced Zoning support flags are shown in table 310.

Table 310 – Switch Enhanced Zoning support flags

Bit Position	Description
0	Enhanced Zoning supported. When this bit is set to one, the Switch is able to support Enhanced Zoning mode. When this bit is set to zero, the Switch is not able to support Enhanced Zoning mode.
1	Enhanced Zoning enabled. When this bit is set to one, the Switch is working in Enhanced Zoning mode. When this bit is set to zero, the Switch is working in Basic Zoning mode.
2	Merge Control Setting. When this bit is set to one, this Switch is working in Restrict mode, so it may join a Fabric only if the Fabric's Zoning Database is equal to its Zoning Database. When this bit is set to zero, this Switch is working in Allow mode, so it may join a Fabric only if the Fabric's Zoning Database is able to merge with its Zoning Database.
3	Default Zone Setting. When this bit is set to one this Switch denies traffic between members of the Default Zone. When this bit is set to zero this Switch permit traffic between members of the Default Zone.
4	Zone Set Database supported. When this bit is set to one, the Zone Server on this Switch is able to maintain a Zone Set Database. When this bit is set to zero, the Zone Server on this Switch is not able to maintain a Zone Set Database.
5	Zone Set Database enabled. When this bit is set to one, the Zone Server on this Switch is maintaining a Zone Set Database. When this bit is set to zero, the Zone Server on this Switch is not maintaining a Zone Set Database.
6	Activate Direct command supported. When this bit is set to one, this Switch supports the Activate Direct command. When this bit is set to zero, this Switch does not support the Activate Direct command.
7	Hard Zoning supported. When this bit is set to one, this Switch supports Hard Zoning. When this bit is set to zero, this Switch does not support Hard Zoning.
8	FC-SP Zoning Supported (see FC-SP-2).
9	FC-SP Zoning Enabled (see FC-SP-2).
10 to 31	Reserved

6.4.10.1.2 Set Fabric Enhanced Zoning Support (SFEZ)

The Fabric Zone Server shall, if it receives a SFEZ request, set the Fabric to work in Enhanced Zoning mode. If the Fabric is not capable of working in Enhanced Zoning, then the Fabric Zone Server shall reject the request with reason code explanation “Fabric not able to work in Enhanced Mode”.

The SFEZ request does not require a Server Session. When the Fabric is working in Enhanced mode the SFEZ request may be used to modify the Fabric Zoning policies. When the Fabric is working in enhanced mode it is not possible to revert to Basic mode.

The format of the GFEZ Request CT_IU is shown in table 311.

Table 311 – SFEZ Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Fabric Enhanced Zoning request flags	4

The Fabric Enhanced Zoning request flags are shown in table 312.

Table 312 – Fabric Enhanced Zoning request flags

Bit Position	Description
0	Reserved
1	Enhanced Zoning enable. When this bit is one, the managing application is requesting the Fabric to set its Zoning operational mode as Enhanced. When this bit is zero, the managing application is requesting the Fabric to not change its Zoning operational mode.
2	Merge Control Setting. When this bit is one, the managing application is requesting the Fabric to operate in Restrict mode. When this bit is zero, the managing application is requesting the Fabric to operate in Allow mode.
3	Default Zone Setting. When this bit is one, the managing application is requesting the Fabric to deny traffic and visibility among members of the Default Zone. When this bit is zero, the managing application is requesting the Fabric to allow traffic and visibility among members of the Default Zone.
4	FC-SP Zoning Enable (see FC-SP-2).
5 to 31	Reserved

The format of the Accept CT_IU to a SFEZ request is shown in table 313.

Table 313 – Accept CT_IU to SFEZ Request

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.4.10.2 Session Commands

6.4.10.2.1 Server Session Begin (SSB)

See 4.9.5.2.

6.4.10.2.2 Server Session End (SSE)

See 4.9.5.3.

6.4.10.2.3 Commit Zone Changes (CMIT)

The Fabric Zone Server shall, if it receives a CMIT operation request, commit all outstanding modifications made by the issuing Client to the Zone Set Database.

The CMIT processing may persist longer than the Common Transport timeout (i.e., 3*R_A_TOV). The Fabric Zone Server shall reply within the Common Transport timeout. The reply shall be a Reject CT_IU having reason code ‘Logical Busy’ and reason code explanation ‘Request in Process’ until the CMIT processing completes successfully or unsuccessfully. The Fabric Zone Server shall return a response to the CMIT Request other than a Reject CT_IU with a ‘Logical Busy’ reason code and a ‘Request in Process’ reason code explanation when the CMIT processing completes. The Fabric Zone Server shall respond to any other Fabric Zone Server Requests with a Reject CT_IU with a ‘Logical Busy’ reason code and a ‘CMIT in Process’ reason code explanation until a response to the CMIT Request other than a Reject CT_IU with a ‘Logical Busy’ reason code and a ‘Request in Process’ reason code explanation has been sent.

NOTE 41 – The management application should retransmit the CMIT Request until an Accept CT_IU or a Reject CT_IU with a reason code other than ‘Logical Busy’ and reason code explanation other than ‘Request in Process’ is received, or until the application gets tired of it. When the Fabric is processing a CMIT Request, any subsequently received CMIT Requests do not interrupt or restart the processing in progress. Instead, subsequent CMIT Requests are a way for the management application to know when and how the CMIT processing completes.

The CMIT Request shall not be used in Basic Zoning.

The format of the CMIT Request payload is depicted in table 314 below.

Table 314 – CMIT Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3

The format of the CMIT Accept payload is depicted in table 315 below:

Table 315 – CMIT Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.4.10.3 Basic Zoning Commands

6.4.10.3.1 Query - Get Capabilities (GZC)

The Fabric Zone Server shall, if it receives a GZC request, return the Zone capabilities supported by the Fabric. The format of the GZC Request CT_IU is shown in table 316.

Table 316 – GZC Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

The format of the Accept CT_IU to a GZC request is shown in table 317.

Table 317 – Accept CT_IU to GZC Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Capability flags	1
Reserved	3
Vendor specific capabilities	4

The capability flags indicate the Zone capabilities supported by the Fabric, as shown in table 318.

Table 318 – Capability flags

Bit Position	Description
7	Soft Zones supported. When this bit is one, the Fabric supports Soft Zones.
6	Hard Zones supported. When this bit is one, the Fabric supports Hard Zones.
5 to 1	Reserved
0	Zone Set Database available. When this bit is zero, the Fabric Zone Server does not maintain a Zone Set Database (i.e., the only way to activate a Zone Set is via the AZSD request, see 6.4.10.3.10). When this bit is one, the Fabric Zone Server maintains a Zone Set Database.

The vendor specific capabilities field is not defined by this standard.

6.4.10.3.2 Query - Get Enforcement State (GEST)

The Fabric Zone Server shall, if it receives a GEST request, return the current capability being enforced by the Fabric. The method by which this capability is selected is not defined by this standard. The format of the GEST Request CT_IU is shown in table 319.

Table 319 – GEST Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

The format of the Accept CT_IU to a GEST request is shown in table 320.

Table 320 – Accept CT_IU to GEST Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone state	1
Reserved	3
Vendor specific state	4

The Zone state indicates whether the Fabric has an Active Zone Set, and the current enforcement of the Zone Set, as shown in table 321.

Table 321 – Enforcement state flags

Bit Position	Description
7	Soft Zone Set Enforced. When this bit is one, the Fabric is currently enforcing a Soft Zone Set.
6	Hard Zone Set Enforced. When this bit is one, the Fabric is currently enforcing a Hard Zone Set.
5 to 0	Reserved

The vendor specific state field is not defined by this standard.

6.4.10.3.3 Query - Get Zone Set List (GZSN)

The Fabric Zone Server shall, if it receives a GZSN request, return the Zone Set attributes of all Zone Sets in the Zone Set Database. The format of the GZSN Request CT_IU is shown in table 322.

Table 322 – GZSN Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

The format of the Accept CT_IU to a GZSN request is shown in table 323.

Table 323 – Accept CT_IU to GZSN Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Number of Zone Set attribute entries (n)	4
Zone Set Name #1	see 6.4.8.1
Number of Zones #1	4
Zone Set Name #2	see 6.4.8.1
Number of Zones #2	4
...	
Zone Set Name #n	see 6.4.8.1
Number of Zones #n	4

One or more Zone Set attribute entries are returned, and the Zone Set attribute entries may be returned in any order, and the order may be different for every request even if the same Zone Set attribute entries are returned and the requestor is the same.

6.4.10.3.4 Query - Get Zone List (GZD)

The Fabric Zone Server shall, if it receives a GZD request, return the Zone attributes of all Zones in the specified Zone Set. The format of the GZD Request CT_IU is shown in table 324.

Table 324 – GZD Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Set Name	8 to 68

The format of the Accept CT_IU to a GZD request is shown in table 325.

Table 325 – Accept CT_IU to GZD Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Number of Zone attribute entries (n)	4
Zone Name #1	see 6.4.8.1
Number of Zone Members #1	4
Zone Name #2	see 6.4.8.1
Number of Zone Members #2	4
...	
Zone Name #n	see 6.4.8.1
Number of Zone Members #n	4

One or more Zone attribute entries are returned, and the Zone attribute entries may be returned in any order, and the order may be different for every request even if the same Zone attribute entries are returned and the requestor is the same.

6.4.10.3.5 Query - Get Zone Member List (GZM)

The Fabric Zone Server shall, if it receives a GZM request, return the Zone Member attributes of all Zone Members in the specified Zone. The format of the GZM Request CT_IU is shown in table 326.

Table 326 – GZM Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Name	see 6.4.8.1

The format of the Accept CT_IU to a GZM request is shown in table 327.

Table 327 – Accept CT_IU to GZM Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Number of Zone Member attribute entries (n)	4
Zone Member Identifier Type #1	1
Reserved	3
Zone Member Identifier #1	8
Zone Member Identifier Type #2	1
Reserved	3
Zone Member Identifier #2	8
...	
Zone Member Identifier Type #n	1
Reserved	3
Zone Member Identifier #n	8

One or more Zone Member attribute entries are returned, and the Zone Member attribute entries may be returned in any order, and the order may be different for every request even if the same Zone Member attribute entries are returned and the requestor is the same.

6.4.10.3.6 Query - Get Active Zone Set (GAZS)

The Fabric Zone Server shall, if it receives a GAZS request, return the Zone Set attributes of the Active Zone Set. The format of the GAZS Request CT_IU is shown in table 328.

Table 328 – GAZS Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

The format of the Accept CT_IU to a GAZS request is shown in table 329.

Table 329 – Accept CT_IU to GAZS Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Set Name of Active Zone Set	see 6.4.8.1
Number of Zones	4
Zone Name #1	see 6.4.8.1
Number of Zone Members #1	4
Zone Member Identifier Type #1	1
Reserved	3
Zone Member Identifier #1	8
Zone Member Identifier Type #2	1
Reserved	3
Zone Member Identifier #2	8
...	
Zone Member Identifier Type #m	1
Reserved	3
Zone Member Identifier #m	8
Zone Name #2	see 6.4.8.1

Table 329 – Accept CT_IU to GAZS Request (Continued)

Item	Size (Bytes)
Number of Zone Members #2	4
Zone Member Identifier Type #1	1
Reserved	3
Zone Member Identifier #1	8
Zone Member Identifier Type #2	1
Reserved	3
Zone Member Identifier #2	8
...	
Zone Member Identifier Type #m	1
Reserved	3
Zone Member Identifier #m	8
...	
Zone Name #n	see 6.4.8.1
Number of Zone Members #n	4
Zone Member Identifier Type #1	1
Reserved	3
Zone Member Identifier #1	8
Zone Member Identifier Type #2	1
Reserved	3
Zone Member Identifier #2	8
...	
Zone Member Identifier Type #m	1
Reserved	3
Zone Member Identifier #m	8

6.4.10.3.7 Query - Get Zone Set (GZS)

The Fabric Zone Server shall, if it receives a GZS request, return the Zone Set attributes of the specified Zone Set. The format of the GZS Request CT_IU is shown in table 330.

Table 330 – GZS Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Set Name	see 6.4.8.1

The format of the Accept CT_IU to a GZS request is shown in table 331.

Table 331 – Accept CT_IU to GZS Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Set Name of Zone Set	see 6.4.8.1
Number of Zones	4
Zone Name #1	see 6.4.8.1
Number of Zone Members #1	4
Zone Member Identifier Type #1	1
Reserved	3
Zone Member Identifier #1	8
Zone Member Identifier Type #2	1
Reserved	3
Zone Member Identifier #2	8
...	
Zone Member Identifier Type #m	1
Reserved	3
Zone Member Identifier #m	8

Table 331 – Accept CT_IU to GZS Request (Continued)

Item	Size (Bytes)
Zone Name #2	see 6.4.8.1
Number of Zone Members #2	4
Zone Member Identifier Type #1	1
Reserved	3
Zone Member Identifier #1	8
Zone Member Identifier Type #2	1
Reserved	3
Zone Member Identifier #2	8
...	
Zone Member Identifier Type #m	1
Reserved	3
Zone Member Identifier #m	8
...	
Zone Name #n	see 6.4.8.1
Number of Zone Members #n	4
Zone Member Identifier Type #1	1
Reserved	3
Zone Member Identifier #1	8
Zone Member Identifier Type #2	1
Reserved	3
Zone Member Identifier #2	8
...	
Zone Member Identifier Type #m	1
Reserved	3

Table 331 – Accept CT_IU to GZS Request (Continued)

Item	Size (Bytes)
Zone Member Identifier #m	8

6.4.10.3.8 Get Activation Results (GAR)

The Fabric Zone Server shall, if it receives a GAR request, return the results of the last Zone Set activation request. The Fabric Zone Server is only required to keep state information of the last activation request. Since Fabric Zone Server requests support multiple Client requests, use of a serialization mechanism such as Server Session Begin/End ensures that the GAR response is associated with the Client's activation request.

NOTE 42 – It is recommended that a Fabric operating in basic Zoning use SSB and SSE as a serialization mechanism.

The format of the GAR Request CT_IU is shown in table 332.

Table 332 – GAR Request CT_IU

Item	Size (bytes)
CT_IU Preamble	see 4.3

The format of the Accept CT_IU to a GAR request is shown in table 333.

Table 333 – GAR Accept CT_IU

Item	Size (bytes)
CT_IU Preamble	see 4.3
Number of Interconnect Element entries (n)	4
Interconnect Element Name #1	8
Reserved	3
Interconnect Element #1 Domain ID	1
Reserved	3
Interconnect Element #1 Reason Code	1
Interconnect Element Name #2	8
Reserved	3
Interconnect Element #2 Domain ID	1
Reserved	3
Interconnect Element #2 Reason Code	1

Interconnect Element Name #n	8
Reserved	3
Interconnect Element #n Domain ID	1
Reserved	3
Interconnect Element #n Reason Code	1

The reason code for one or more Interconnect elements' information is returned.

For each Switch, the Interconnect Element name, domain id and reason code are returned. If the activation request was successful, an Accept CT_IU is returned with Number of Interconnect Element entries set to 0 (i.e., an empty list).

The Interconnect Element Reason Code shall be as specified in FC-SW-6.

6.4.10.3.9 Add Zone Set (ADZS)

The ADZS Fabric Zone Server request shall be used to create a new Zone Set or to replace an existing Zone Set.

The Zone Set Name field of the Request CT_IU may be equal to a currently defined Zone Set Name. If the value of the Zone Set Name is not equal to a currently defined Zone Set Name, the Fabric Zone Server shall create a new Zone Set object and add the specified Zones and Zone Members to the new Zone Set. If the value of the Zone Set Name is equal to a currently defined Zone Set Name, the Fabric Zone Server shall remove the existing Zone Set prior to creating the new Zone Set object.

The format of the ADZS Request CT_IU is shown in table 334.

Table 334 – ADZS Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Set Name	see 6.4.8.1
Number of Zones	4
Zone Name #1	see 6.4.8.1
Number of Zone Members #1	4
Zone Member Identifier Type #1	1
Reserved	3
Zone Member Identifier #1	8
Zone Member Identifier Type #2	1
Reserved	3
Zone Member Identifier #2	8
...	
Zone Member Identifier Type #m	1
Reserved	3
Zone Member Identifier #m	8
Zone Name #2	see 6.4.8.1
Number of Zone Members #2	4
Zone Member Identifier Type #1	1
Reserved	3
Zone Member Identifier #1	8

Table 334 – ADZS Request CT_IU (Continued)

Item	Size (Bytes)
Zone Member Identifier Type #2	1
Reserved	3
Zone Member Identifier #2	8
...	
Zone Member Identifier Type #m	1
Reserved	3
Zone Member Identifier #m	8
...	
Zone Name #n	see 6.4.8.1
Number of Zone Members #n	4
Zone Member Identifier Type #1	1
Reserved	3
Zone Member Identifier #1	8
Zone Member Identifier Type #2	1
Reserved	3
Zone Member Identifier #2	8
...	
Zone Member Identifier Type #m	1
Reserved	3
Zone Member Identifier #m	8

The format of the ADZS Accept CT_IU is shown in table 335.

Table 335 – ADZS Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.4.10.3.10 Activate Zone Set Direct (AZSD)

The AZSD Fabric Zone Server request shall be used to activate the specified Zone Set. If there is currently an Active Zone Set, it shall be deactivated before the specified Zone Set is activated.

The Zone Set defined in the Request CT_IU shall not be added to the Zone Set Database.

The format of the AZSD Request CT_IU is shown in table 336.

Table 336 – AZSD Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Set Name	see 6.4.8.1
Number of Zones	4
Zone Name #1	see 6.4.8.1
Number of Zone Members #1	4
Zone Member Identifier Type #1	1
Reserved	3
Zone Member Identifier #1	8
Zone Member Identifier Type #2	1
Reserved	3
Zone Member Identifier #2	8
...	
Zone Member Identifier Type #m	1
Reserved	3
Zone Member Identifier #m	8
Zone Name #2	see 6.4.8.1
Number of Zone Members #2	4
Zone Member Identifier Type #1	1
Reserved	3

Table 336 – AZSD Request CT_IU (Continued)

Item	Size (Bytes)
Zone Member Identifier #1	8
Zone Member Identifier Type #2	1
Reserved	3
Zone Member Identifier #2	8
...	
Zone Member Identifier Type #m	1
Reserved	3
Zone Member Identifier #m	8
...	
Zone Name #n	see 6.4.8.1
Number of Zone Members #n	4
Zone Member Identifier Type #1	1
Reserved	3
Zone Member Identifier #1	8
Zone Member Identifier Type #2	1
Reserved	3
Zone Member Identifier #2	8
...	
Zone Member Identifier Type #m	1
Reserved	3
Zone Member Identifier #m	8

The format of the AZSD Accept CT_IU is shown in table 337.

Table 337 – AZSD Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.4.10.3.11 Activate Zone Set (AZS)

The Fabric Zone Server shall, if it receives an AZS request, activate the specified Zone Set. If the value of the Zone Set Name is not equal to a currently defined Zone Set Name, the Fabric Zone Server shall reject the AZS request. The format of the AZS Request CT_IU is shown in table 338.

Table 338 – AZS Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Set Name to Activate	see 6.4.8.1

The format of the Accept CT_IU to an AZS request is shown in table 339.

Table 339 – Accept CT_IU to AZS Request

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.4.10.3.12 Deactivate Zone Set (DZS)

The Fabric Zone Server shall, if it receives a DZS request, deactivate the current Active Zone Set. The format of the DZS Request CT_IU is shown in table 340.

Table 340 – DZS Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

The format of the Accept CT_IU to a DZS request is shown in table 341.

Table 341 – Accept CT_IU to DZS Request

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.4.10.3.13 Add Zone Members (AZM)

The AZM Fabric Zone Server request shall be used to add one or more Zone Members to an existing Zone.

The format of the AZM Request CT_IU is shown in table 342.

Table 342 – AZM Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Name	see 6.4.8.1
Zone Member Identifier Type #1	1
Reserved	3
Zone Member Identifier #1	8
Zone Member Identifier Type #2	1
Reserved	3
Zone Member Identifier #2	8
...	
Zone Member Identifier Type #m	1
Reserved	3
Zone Member Identifier #m	8

The format of the AZM Accept CT_IU is shown in table 343.

Table 343 – AZM Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.4.10.3.14 Add Zone (AZD)

The AZD Fabric Zone Server request shall be used to add a Zone to an existing Zone Set.

The format of the AZD Request CT_IU is shown in table 344.

Table 344 – AZD Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Set Name	see 6.4.8.1
Zone Name	see 6.4.8.1

The format of the AZD Accept CT_IU is shown in table 345.

Table 345 – AZD Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.4.10.3.15 Remove Zone Members (RZM)

The RZM Fabric Zone Server request shall be used to remove one or more Zone Members from an existing Zone.

The format of the RZM Request CT_IU is show in table 346.

.

Table 346 – RZM Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Name	see 6.4.8.1
Zone Member Identifier Type #1	1
Reserved	3
Zone Member Identifier #1	8
Zone Member Identifier Type #2	1
Reserved	3
Zone Member Identifier #2	8
...	
Zone Member Identifier Type #m	1
Reserved	3
Zone Member Identifier #m	8

The format of the RZM Accept CT_IU is shown in table 347.

Table 347 – RZM Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.4.10.3.16 Remove Zone (RZD)

The RZD Fabric Zone Server request shall be used to remove a Zone from an existing Zone Set.

The format of the RZD Request CT_IU is shown in table 348.

Table 348 – RZD Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Set Name	see 6.4.8.1
Zone Name	see 6.4.8.1

The format of the RZD Accept CT_IU is shown in table 349.

Table 349 – RZD Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.4.10.3.17 Remove Zone Set (RZS)

The RZS Fabric Zone Server request shall be used to remove a Zone Set.

The format of the RZS Request CT_IU is shown in table 350.

Table 350 – RZS Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Set Name	see 6.4.8.1

The format of the RZS Accept CT_IU is shown in table 351.

Table 351 – RZS Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.4.10.4 Enhanced Zoning Commands

6.4.10.4.1 Get Activation Results (GAR)

See 6.4.10.3.8.

6.4.10.4.2 Get Zone Attribute Object Name (GZA) Operation

The Fabric Zone Server shall, if it receives a GZA operation request, return the Zone Attribute Object Name for the specified Zone as defined in the Zone Set Database. The GZA request payload shall specify the Zone name identifying the Zone for which the Zone Attribute Object Name is sought. The GZA accept payload contains the Zone Attribute Object Name for the Zone specified in the request.

If the specified Zone does not exist, then the request is rejected with the appropriate reason code explanation.

The format of the GZA request payload is depicted in table 352 below.

Table 352 – GZA Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Name	see 6.4.8.1

The format of the GZA accept payload is depicted in table 353 below

Table 353 – GZA Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Attribute Object Name	see 6.4.8.1

6.4.10.4.3 Get Zone Attribute Block (GZAB) Operation

The Fabric Zone Server shall, if it receives a GZAB operation request, return the Zone Attribute Block associated with the specified Zone Attribute Object as defined in the Zone Set Database. The GZAB request payload shall specify the Zone Attribute Object Name identifying the Zone Attribute Object for which the Zone Attribute Block is sought. The GZAB accept payload contains the Zone Attribute Block for the Zone Attribute Object specified in the request.

If the specified Zone Attribute Object does not exist, then the request is rejected with the appropriate reason code explanation.

The format of the GZAB request payload is depicted in table 354 below.

Table 354 – GZAB Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Attribute Object Name	see 6.4.8.1

The format of the GZAB accept payload is depicted in table 355 below

Table 355 – GZAB Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Attribute Block	see 6.4.8.3.8

The format of the Zone Attribute Block is described in 6.4.8.3.8.

6.4.10.4.4 Get Zone Set List - Enhanced (GZSE) Operation

The Fabric Zone Server shall, if it receives a GZSE operation request, return a list of Zone Sets that are currently defined in the Zone Set Database. The GZSE accept payload shall contain a list of Zone Sets defined in the Zone Set Database.

If no Zone Sets are currently defined in the Zone Set Database, then the number of Zone Set entries is set to zero in the response.

The format of the GZSE request payload is shown in table 356 below.

Table 356 – GZSE Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3

The format of the GZSE accept payload is depicted in table 357 below.

Table 357 – GZSE Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Number of Zone Set Names (n)	4
Zone Set Name 1	see 6.4.8.1
Zone Set Name 2	see 6.4.8.1
...	
Zone Set Name n	see 6.4.8.1

6.4.10.4.5 Get Zone List - Enhanced (GZDE) Operation

The Fabric Zone Server shall, if it receives a GZDE operation request, return a list of Zone references that are currently defined in the Zone Set Database for the specified Zone Set. The GZDE request payload shall contain the name of the Zone Set for which the Zone references are sought. The GZDE accept payload shall contain a list of Zone names that represent Zone references for the specified Zone Set.

If no Zone references are currently defined for the specified Zone Set, then the number of Zone names is set to zero in the response.

If the specified Zone Set does not exist, the command shall be rejected with a reason code of ‘Unable to perform command request’, and additional reason explanation of “Zone Set Unknown”.

The format of the GZDE request payload is shown in table 358 below.

Table 358 – GZDE Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Set Name	see 6.4.8.1

The format of the GZDE accept payload is depicted in table 359 below.

Table 359 – GZDE Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Number of Zone Names (n)	4
Zone Name 1	see 6.4.8.1
Zone Name 2	see 6.4.8.1
...	
Zone Name n	see 6.4.8.1

6.4.10.4.6 Get Zone Member List - Enhanced (GZME) Operation

The Fabric Zone Server shall, if it receives a GZME operation request, return a list of Zone Members for the specified Zone in the Zone Set Database. The GZME request payload shall specify the Zone Name for which the list of Zone Members are sought. The GZME accept payload shall contain a list of Zone Members belonging to the specified Zone.

If no Zone Members are currently registered for the specified Zone, then the number of Zone Member Entries is set to zero in the response. The format of the GZME request payload is depicted in table 360 below.

Table 360 – GZME Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Name	see 6.4.8.1

The format of the GZME accept payload is depicted in table 361 below:

Table 361 – GZME Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Number of Zone Member Entries (n)	4
Zone Member Entry 1	see 6.4.8.2.4
Zone Member Entry 2	see 6.4.8.2.4
...	
Zone Member Entry n	see 6.4.8.2.4

The format of the Zone Member Entry is depicted in table 291.

6.4.10.4.7 Get Zone Attribute Object List (GZAL) Operation

The Fabric Zone Server shall, if it receives a GZAL operation request, return a list of Zone Attribute Objects that are currently defined in the Zone Set Database. The GZAL accept payload shall contain a list of Zone Attribute Objects defined in the Zone Set Database.

If no Zone Attribute Objects are currently defined in the Zone Set Database, then the number of Zone Attribute Objects entries is set to zero in the response.

The format of the GZAL request payload is shown in table 362 below.

Table 362 – GZAL Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3

The format of the GZAL accept payload is depicted in table 363 below.

Table 363 – GZAL Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Number of Zone Attribute Object Names (n)	4
Zone Attribute Object Name 1	see 6.4.8.1
Zone Attribute Object Name 2	see 6.4.8.1
...	
Zone Attribute Object Name n	see 6.4.8.1

6.4.10.4.8 Get Alias List (GAL) Operation

The Fabric Zone Server shall, if it receives a GAL operation request, return a list of Zone Aliases defined in the Zone Set Database. The GAL request payload shall not specify any parameters. The GAL accept payload shall contain a list of Alias names in the Zone Set Database.

If no Aliases are currently defined in the Zone Set Database, then the number of Alias names is set to zero in the response. The format of the GAL Request payload is depicted in table 364 below:

Table 364 – GAL Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3

The format of the GAL Accept payload is depicted in table 365 below:

Table 365 – GAL Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Number of Alias names (n)	4
Alias Name 1	see 6.4.8.1
Alias Name 2	see 6.4.8.1
...	
Alias Name n	see 6.4.8.1

The format of the Alias Name is described in 6.4.8.1.

6.4.10.4.9 Get Alias Member List (GAM) Operation

The Fabric Zone Server shall, if it receives a GAM operation request, return a list of Alias Members for the specified Alias defined in the Zone Set Database. The GAM request payload shall specify the Alias Name for which the list of alias members are sought. The GAM accept payload shall contain a list of alias members belonging to the specified Alias.

If no Alias Members are currently registered for the specified Alias Name, then the number of Alias Member Entries is set to zero in the response. The format of the GAM Request payload is depicted in table 366 below:

Table 366 – GAM Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Alias Name	see 6.4.8.1

The format of the Alias Name is described in 6.4.8.1.

The format of the GAM Accept payload is depicted in table 367 below:

Table 367 – GAM Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Number of Alias Member Entries (n)	4
Alias Member Entry 1	see 6.4.8.3.7
Alias Member Entry 2	see 6.4.8.3.7
...	
Alias Member Entry n	see 6.4.8.3.7

The Alias Member Entry uses the Zone Member Object format depicted in table 291 with the Zone Member Identifier Type field set to 05h.

6.4.10.4.10 Set Zone Attribute Object Name (SZA) Operation

The Fabric Zone Server shall, if it receives a SZA operation request, set the specified Zone Attribute Object Name for the specified Zone in the Zone Set Database. The SZA request payload shall specify the Zone Name identifying the Zone for which the Zone Attribute Object Name is to be set.

If the specified Zone does not exist, then the request is rejected with the appropriate reason code explanation.

The format of the SZA Request payload is depicted in table 368 below.

Table 368 – SZA Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Name	see 6.4.8.1
Zone Attribute Object Name	see 6.4.8.1

The format of the SZA Accept payload is depicted in table 369 below

Table 369 – SZA Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.4.10.4.11 Set Zone Attribute Block (SZAB) Operation

The Fabric Zone Server shall, if it receives a SZAB operation request, set the specified attributes for the specified Zone Attribute Object in the Zone Set Database. The SZAB request payload shall specify the Zone Attribute Object Name identifying the Zone Attribute Object for which the attributes are to be set.

If the specified Zone Attribute Object does not exist, then the request is rejected with the appropriate reason code explanation.

The format of the SZAB Request payload is depicted in table 370 below.

Table 370 – SZAB Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Attribute Object Name	see 6.4.8.1
Zone Attribute Block	see 6.4.8.3.8

The format of the Zone Attribute Block is described in 6.4.8.3.8.

The format of the SZAB Accept payload is depicted in table 371 below

Table 371 – SZAB Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.4.10.4.12 Create Zone Set (CZS) Operation

The Fabric Zone Server shall, if it receives a CZS operation request, create a Zone Set in the Zone Set Database that references the Zones identified by the specified Zone Names. The CZS request payload shall specify the Zone Set Name and zero or more Zone Names.

If no Zone Names are specified for the Zone Set, then the Zone Set is created and contains a zero number of Zones. The specified zones may not exist at the time of the CZS request.

NOTE 43 – If all zones in a Zone Set are not created by the time a CMIT request is sent, the CMIT fails.

The format of the CZS Request payload is depicted in table 372 below.

Table 372 – CZS Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Set Name	see 6.4.8.1
Number of Zone Names (n)	4
Zone Name 1	see 6.4.8.1
Zone Name 2	see 6.4.8.1
...	
Zone Name n	see 6.4.8.1

Number of Zone Names: This specifies the number of Zone Names contained in the payload.

Zone Name: Specifies the name of a Zone that is a member of the Zone Set.

The format of the CZS Accept payload is depicted in table 373 below:

Table 373 – CZS Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.4.10.4.13 Create Zone (CZ) Operation

The Fabric Zone Server shall, if it receives a CZ operation request, create a Zone in the Zone Set Database that contains the specified Zone Members. The CZ request payload shall specify the Zone Name and zero or more Zone Members.

If no Zone Members are specified for the Zone, then the Zone is created and contains a zero number of Zone Members. The format of the CZ Request payload is depicted in table 374 below.

Table 374 – CZ Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Name	see 6.4.8.1
Number of Zone Members (n)	4
Zone Member Entry 1	see 6.4.8.2.4
Zone Member Entry 2	see 6.4.8.2.4
...	
Zone Member Entry n	see 6.4.8.2.4

Number of Zone Members: This specifies the number of Zone Members contained in the payload.

Zone Member Entry: Specifies a Zone Member that is a member of the Zone. The Zone Member Entry uses the Zone Member Object format depicted in table 291.

The format of the CZ Accept payload is depicted in table 375 below:

Table 375 – CZ Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.4.10.4.14 Create Alias (CA) Operation

The Fabric Zone Server shall, if it receives a CA operation request, create an Alias in the Zone Set Database that contains the specified Alias Members. The CA request payload shall specify the Alias Name and zero or more Alias Members.

If no Alias Members are specified for the Alias, then the Alias is created and contains a zero number of Alias Members. The format of the CA Request payload is depicted in table 376 below.

Table 376 – CA Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Alias Name	see 6.4.8.1
Number of Alias Members (n)	4
Alias Member Entry 1	see 6.4.8.3.7
Alias Member Entry 2	see 6.4.8.3.7
...	
Alias Member Entry n	see 6.4.8.3.7

Alias Name: The format of the Alias Name is described in 6.4.8.1.

Number of Alias Members: This specifies the number of Alias Members contained in the payload.

Alias Member Entry: Specifies an Alias Member that is a member of the Alias. The Alias Member Entry uses the Zone Member Object format depicted in table 291 with the Zone Member Identifier Type field set to 05h.

The format of the CA Accept payload is depicted in table 377 below:

Table 377 – CA Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.4.10.4.15 Create Zone Attribute Object (CZA) Operation

The Fabric Zone Server shall, if it receives a CZA operation request, create an Zone Attribute Object in the Zone Set Database that contains the specified Zone Attribute Block. The CZA request payload shall specify the Zone Attribute Object Name and the Zone Attribute Block.

The format of the CZA Request payload is depicted in table 378 below.

Table 378 – CZA Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Attribute Object Name	see 6.4.8.1
Zone Attribute Block	see 6.4.8.3.8

Zone Attribute Object Name: The format of the Zone Attribute Object Name is described in 6.4.8.1.

Zone Attribute Block: The format of the Zone Attribute Block is described in 6.4.8.3.8.

The format of the CA Accept payload is depicted in table 379 below:

Table 379 – CZA Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.4.10.4.16 Add Zones Enhanced (AZ) Operation

The Fabric Zone Server shall, if it receives an AZ operation request, add one or more Zone references to the specified Zone Set in the Zone Set Database. The AZ request payload shall specify the Zone Set Name for which the specified list of Zone Names are to be added.

If the specified Zone Set does not exist, then the request is rejected with the appropriate reason code explanation.

The format of the AZ Request payload is depicted in table 380 below.

Table 380 – AZ Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Set Name	see 6.4.8.1
Number of Zone Names (n)	4
Zone Name 1	see 6.4.8.1
Zone Name 2	see 6.4.8.1
...	
Zone Name n	see 6.4.8.1

Number of Zone Names: This specifies the number of Zone Names contained in the payload.

Zone Name: Specifies the name of a Zone that is a member of the Zone Set.

The format of the AZ Accept payload is depicted in table 381 below:

Table 381 – AZ Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.4.10.4.17 Remove Zones (RZ) Operation

The Fabric Zone Server shall, if it receives a RZ operation request, remove one or more Zone References from the specified Zone Set in the Zone Set Database. The RZ request payload shall specify the Zone Set Name for which the list of Zone Names are to be removed.

If the specified Zone Set does not exist, then the request is rejected with the appropriate reason code explanation. If a specified Zone is not a member of the Zone Set this shall not constitute an error.

The format of the RZ Request payload is depicted in table 382 below.

Table 382 – RZ Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Set Name	see 6.4.8.1
Number of Zone Names (n)	4
Zone Name 1	see 6.4.8.1
Zone Name 2	see 6.4.8.1
...	
Zone Name n	see 6.4.8.1

Number of Zone Names: This specifies the number of Zone Names contained in the payload.

Zone Name: Specifies the name of a Zone that is a member of the Zone Set.

The format of the RZ Accept payload is depicted in table 383 below:

Table 383 – RZ Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.4.10.4.18 Add Zone Members - Enhanced (AZME) Operation

The Fabric Zone Server shall, if it receives a AZME operation request, add one or more Zone Members to an existing Zone defined in the Zone Set Database. The AZME request payload shall specify the Zone Name for which the list of Zone Members are to be added.

If a specified Zone Member is already a member of the Zone, this shall not constitute an error.

The format of the AZME Request payload is depicted in table 384 below.

Table 384 – AZME Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Name	see 6.4.8.1
Number of Zone Member Entries (n)	4
Zone Member Entry 1	see 6.4.8.2.4
Zone Member Entry 2	see 6.4.8.2.4
...	
Zone Member Entry n	see 6.4.8.2.4

The Zone Member Entry uses the Zone Member Object format depicted in table 291.

The format of the AZME Accept payload is depicted in table 385 below:

Table 385 – AZME Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.4.10.4.19 Remove Zone Members - Enhanced (RZME) Operation

The Fabric Zone Server shall, if it receives a RZME operation request, removes one or more Zone Members from an existing Zone in the Zone Set Database. The RZME request payload shall specify the Zone Name for which the list of Zone Members are to be removed.

If the specified Zone does not exist, then the request is rejected with the appropriate reason code explanation. If a specified Zone Member is not a member of the Zone this shall not constitute an error.

The format of the AZME Request payload is depicted in table 386 below.

Table 386 – RZME Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Name	see 6.4.8.1
Number of Zone Member Entries (n)	4
Zone Member Entry 1	see 6.4.8.2.4
Zone Member Entry 2	see 6.4.8.2.4
...	
Zone Member Entry n	see 6.4.8.2.4

The Zone Member Entry uses the Zone Member Object format depicted in table 291.

The format of the RZME Accept payload is depicted in table 387 below:

Table 387 – RZME Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.4.10.4.20 Add Alias Members (AAM) Operation

The Fabric Zone Server shall, if it receives a AAM operation request, add one or more members to an existing Alias in the Zone Set Database. The AAM request payload shall specify the Alias Name for which the list of alias members are to be added.

If the Alias specified in the request does not exist, then the request is rejected and the proper reason code explanation is indicated. If a specified Alias Member is already a member of the Alias, this shall not constitute and error.

The format of the AAM Request payload is depicted in table 388 below.

Table 388 – AAM Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Alias Name	see 6.4.8.1
Number of Alias Member Entries (n)	4
Alias Member Entry 1	see 6.4.8.3.7
Alias Member Entry 2	see 6.4.8.3.7
...	
Alias Member Entry n	see 6.4.8.3.7

The format of the Alias Name is described in 6.4.8.1. The Alias Member Entry uses the Zone Member Object format depicted in table 291 with the Zone Member Identifier Type field set to 05h.

The format of the AAM Accept payload is depicted in table 389 below

Table 389 – AAM Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.4.10.4.21 Remove Alias Members (RAM) Operation

The Fabric Zone Server shall, if it receives a RAM operation request, remove the specified Alias Members from an existing Alias in the Zone Set Database. The RAM request payload shall specify the Alias Name for which the list of Alias Members are to be removed.

If the specified Alias does not exist, then the request is rejected with the appropriate reason code explanation. If a specified Alias Member is not a member of the Alias this shall not constitute an error.

The format of the RAM Request payload is depicted in table 390 below.

Table 390 – RAM Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Alias Name	see 6.4.8.1
Number of Alias Member Entries (n)	4
Alias Member Entry 1	see 6.4.8.3.7
Alias Member Entry 2	see 6.4.8.3.7
...	
Alias Member Entry n	see 6.4.8.3.7

Alias Name: The format of the Alias Name is described in 6.4.8.1.

Alias Member Entry: The Alias Member Entry uses the Zone Member Object format depicted in table 291 with the Zone Member Identifier Type field set to 05h.

The format of the RAM Accept payload is depicted in table 391 below:

Table 391 – RAM Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.4.10.4.22 Delete Zone Set (DLZS) Operation

The Fabric Zone Server shall, if it receives a DLZS operation request, remove the specified Zone Set and its members from the Zone Set Database. The DLZS request payload shall specify the Zone Set Name that is to be deleted from the Fabric.

If the specified Zone Set does not exist, then the request is rejected with the appropriate reason code explanation.

The format of the DLZS Request payload is depicted in table 392 below.

Table 392 – DLZS Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Set Name	see 6.4.8.1

The format of the DLZS Accept payload is depicted in table 393 below:

Table 393 – DLZS Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.4.10.4.23 Delete Zone (DLZ) Operation

The Fabric Zone Server shall, if it receives a DLZ operation request, remove the specified Zone and its members from the Zone Set Database. The DLZ request payload shall specify the Zone Name that is to be deleted from the Fabric.

If the specified Zone does not exist, then the request is rejected with the appropriate reason code explanation.

The format of the DLZ Request payload is depicted in table 394 below.

Table 394 – DLZ Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Name	see 6.4.8.1

The format of the DLZ Accept payload is depicted in table 395 below:

Table 395 – DLZS Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.4.10.4.24 Delete Alias (DLA) Operation

The Fabric Zone Server shall, if it receives a DLA operation request, delete the specified Alias and its Members from the Zone Set Database. The DLA request payload shall specify the Alias Name that is to be deleted.

If the specified Alias does not exist, then the request is rejected with the appropriate reason code explanation.

The format of the DLA Request payload is depicted in table 396 below.

Table 396 – DLA Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Alias Name	see 6.4.8.1

Alias Name: The format of the Alias Name is described in 6.4.8.1.

The format of the DLA Accept payload is depicted in table 397 below:

Table 397 – DLA Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.4.10.4.25 Delete Zone Attribute Object (DLZA) Operation

The Fabric Zone Server shall, if it receives a DLZA operation request, delete the specified Zone Attribute Object from the Zone Set Database. The DLZA request payload shall specify the name of the Zone Attribute Object that is to be deleted.

If the specified Zone Attribute Object does not exist, then the request is rejected with the appropriate reason code explanation.

The format of the DLZA Request payload is depicted in table 398 below.

Table 398 – DLZA Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Attribute Object Name	see 6.4.8.1

Zone Attribute Object Name: The format of the Zone Attribute Object Name is described in 6.4.8.1.

The format of the DLA Accept payload is depicted in table 399 below:

Table 399 – DLZA Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.4.10.4.26 Get Active Zone Set - Enhanced (GAZSE) Operation

The Fabric Zone Server shall, if it receives a GAZSE request, return the Zone Set attributes of the Active Zone Set. The format of the GAZSE Request CT_IU is shown in table 400.

Table 400 – GAZSE Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

The format of the Accept CT_IU to a GAZSE request is shown in table 401.

Table 401 – Accept CT_IU to GAZSE Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Number of Zones	4
Zone Name #1	see 6.4.8.1
Number of Zone Attribute Entries #1	4
Zone Attribute Entry #1	see table 300
Zone Attribute Entry #2	see table 300
...	
Zone Attribute Entry #k	see table 300
Number of Zone Members #1	4
Zone Member #1	see 6.4.8.3.6
Zone Member #2	see 6.4.8.3.6
...	
Zone Member #m	see 6.4.8.3.6
Zone Name #2	see 6.4.8.1
Number of Zone Attribute Entries #2	4
Zone Attribute Entry #1	see table 300
Zone Attribute Entry #2	see table 300
...	

Table 401 – Accept CT_IU to GAZSE Request (Continued)

Item	Size (Bytes)
Zone Attribute Entry #k	see table 300
Number of Zone Members #2	4
Zone Member #1	see 6.4.8.3.6
Zone Member #2	see 6.4.8.3.6
...	
Zone Member #m	see 6.4.8.3.6
...	
Zone Name #n	see 6.4.8.1
Number of Zone Attribute Entries #n	4
Zone Attribute Entry #1	see table 300
Zone Attribute Entry #2	see table 300
...	
Zone Attribute Entry #k	see table 300
Number of Zone Members #n	4
Zone Member #1	see 6.4.8.3.6
Zone Member #2	see 6.4.8.3.6
...	
Zone Member #m	see 6.4.8.3.6

The format of the Zone Attribute Entry is defined in 6.4.8.3.8.

The format of the Zone Member is defined in 6.4.8.3.6.

6.4.10.4.27 Activate Zone Set Direct - Enhanced (AZSDE) Operation

The AZSDE Fabric Zone Server request shall be used to activate the specified Zone Set. If there is currently an Active Zone Set, it shall be deactivated before the specified Zone Set is activated.

The format of the AZSDE Request CT_IU is shown in table 402.

Table 402 – AZSDE Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Number of Zones	4
Zone Name #1	see 6.4.8.1
Number of Zone Attribute Entries #1	4
Zone Attribute Entry #1	see table 300
Zone Attribute Entry #2	see table 300
...	
Zone Attribute Entry #k	see table 300
Number of Zone Members #1	4
Zone Member #1	see 6.4.8.3.6
Zone Member #2	see 6.4.8.3.6
...	
Zone Member #m	see 6.4.8.3.6
Zone Name #2	see 6.4.8.1
Number of Zone Attribute Entries #2	4
Zone Attribute Entry #1	see table 300
Zone Attribute Entry #2	see table 300
...	
Zone Attribute Entry #k	see table 300
Number of Zone Members #2	4
Zone Member #1	see 6.4.8.3.6
Zone Member #2	see 6.4.8.3.6
...	
Zone Member #m	see 6.4.8.3.6

Table 402 – AZSDE Request CT_IU (Continued)

Item	Size (Bytes)
...	
Zone Name #n	see 6.4.8.1
Number of Zone Attribute Entries #n	4
Zone Attribute Entry #1	see table 300
Zone Attribute Entry #2	see table 300
...	
Zone Attribute Entry #k	see table 300
Number of Zone Members #n	4
Zone Member #1	see 6.4.8.3.6
Zone Member #2	see 6.4.8.3.6
...	
Zone Member #m	see 6.4.8.3.6

The format of the Zone Attribute Entry is defined in 6.4.8.3.8.

The format of the Zone Member is defined in 6.4.8.3.6.

The format of the AZSDE Accept CT_IU is shown in table 403.

Table 403 – AZSDE Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.4.10.4.28 Activate Zone Set - Enhanced (AZSE) Operation

The Fabric Zone Server shall, if it receives an AZSE request, activate the specified Zone Set. If the value of the Zone Set Name is not equal to a currently defined Zone Set Name, the Fabric Zone Server shall reject the AZSE request. The format of the AZSE Request CT_IU is shown in table 404.

Table 404 – AZSE Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Set Name to Activate	see 6.4.8.1

The format of the Accept CT_IU to an AZSE request is shown in table 405.

Table 405 – Accept CT_IU to AZSE Request

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.4.10.4.29 Deactivate Zone Set - Enhanced (DZSE) Operation

The Fabric Zone Server shall, if it receives a DZSE request, deactivate the current Active Zone Set. The format of the DZSE Request CT_IU is shown in table 406.

Table 406 – DZSE Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

The format of the Accept CT_IU to a DZSE request is shown in table 407.

Table 407 – Accept CT_IU to DZSE Request

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.4.10.4.30 Get Active Peer Zone (GAPZ) Operation

On receiving a GAPZ operation request, the Fabric Zone Server shall perform the following processing:

a) If the Active Zone Set does not contain any Zone having the specified Zone Name, then the GAPZ request shall be rejected with a reason code of ‘Unable to perform command request’ and a reason code explanation of ‘Zone Name unknown’;

b) If the Active Zone Set contains a Zone having the specified Zone Name that is not a Peer Zone, then the GAPZ request shall be rejected with a reason code of ‘Unable to perform command request’ and a reason code explanation of ‘Active Zone is Not a Peer Zone’;

c) If the Active Zone Set contains a Peer Zone having the specified Zone Name, then the GAPZ request shall be accepted and the Fabric Zone Server shall return the Peer Zone definition in the Active Zone Set having the specified Zone Name.

The format of the GAPZ Request CT_IU is shown in table 408.

Table 408 – GAPZ Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Name	see 6.4.8.1

The format of the Accept CT_IU to a GAPZ request is shown in table 409.

Table 409 – Accept CT_IU to GAPZ Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Name	see 6.4.8.1
Number of Zone Attribute Entries (k)	4
Zone Attribute Entry #1	see table 300
Zone Attribute Entry #2	see table 300
...	
Zone Attribute Entry #k	see table 300
Number of Zone Members (m)	4
Zone Member #1	see 6.4.8.3.6
Zone Member #2	see 6.4.8.3.6
...	
Zone Member #m	see 6.4.8.3.6

6.4.10.4.31 Add/Replace Active Peer Zone (AAPZ) Operation

On receiving an AAPZ operation request, the Fabric Zone Server shall perform the following processing:

a) If the provided Zone is not a Peer Zone, then the AAPZ request shall be rejected with a reason code of ‘Unable to perform command request’ and a reason code explanation of ‘Provided Zone is Not a Peer Zone’;

b) If the provided Zone is a Peer Zone and the Active Zone Set contains a Zone having the specified Zone Name that is not a Peer Zone, then the AAPZ request shall be rejected with a reason code of ‘Unable to perform command request’ and a reason code explanation of ‘Active Zone is Not a Peer Zone’;

c) If the provided Zone is a Peer Zone and the Active Zone Set does not contain any Zone having the specified Zone Name, then the AAPZ request shall be accepted and the Fabric Zone Server shall attempt to update the Active Zone Set within a timeout of one minute by adding the specified Peer Zone to the Active Zone Set;

d) If the provided Zone is a Peer Zone and the Active Zone Set contains a Peer Zone having the specified Zone Name, then the AAPZ request shall be accepted and the Fabric Zone Server shall attempt to update the Active Zone Set within a timeout of one minute by replacing the Peer Zone in the Active Zone Set with the provided Peer Zone definition.

NOTE 44 – Updating the Active Zone Set requires to lock the Fabric. The one minute timeout enables the Fabric Zone Server to collect multiple AAPZ and RAPZ requests and coalesce them into a single Fabric Zoning update action.

In order to avoid useless Fabric locks, an implementation should query the Fabric Zone Server with the GAPZ command to verify if a Peer Zone is already in place before issuing an AAPZ command. Normal RSCN processing indicates when a new Peer Zone becomes enforced by the Fabric.

The format of the AAPZ Request CT_IU is shown in table 410.

Table 410 – AAPZ Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Name	see 6.4.8.1
Number of Zone Attribute Entries (k)	4
Zone Attribute Entry #1	see table 300
Zone Attribute Entry #2	see table 300
...	
Zone Attribute Entry #k	see table 300

Table 410 – AAPZ Request CT_IU (Continued)

Item	Size (Bytes)
Number of Zone Members (m)	4
Zone Member #1	see 6.4.8.3.6
Zone Member #2	see 6.4.8.3.6
...	
Zone Member #m	see 6.4.8.3.6

The AAPZ Request CT_IU shall include one Peer Zone Attribute in the Zone Attribute Block.

The format of the Accept CT_IU to a AAPZ request is shown in table 411.

Table 411 – Accept CT_IU to AAPZ Request

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.4.10.4.32 Remove Active Peer Zone (RAPZ) Operation

On receiving an RAPZ operation request, the Fabric Zone Server shall perform the following processing:

a) If the Active Zone Set does not contain any Zone having the specified Zone Name, then the RAPZ request shall be rejected with a reason code of ‘Unable to perform command request’ and a reason code explanation of ‘Zone Name unknown’;

b) If the Active Zone Set contains a Zone having the specified Zone Name that is not a Peer Zone, then the RAPZ request shall be rejected with a reason code of ‘Unable to perform command request’ and a reason code explanation of ‘Active Zone is Not a Peer Zone’;

c) If the Active Zone Set contains a Peer Zone having the specified Zone Name, then the RAPZ request shall be accepted and the Fabric Zone Server shall attempt to update the Active Zone Set within a timeout of one minute by removing the Peer Zone having the specified Zone Name from the Active Zone Set.

NOTE 45 – Updating the Active Zone Set requires to lock the Fabric. The one minute timeout enables the Fabric Zone Server to collect multiple AAPZ and RAPZ requests and coalesce them into a single Fabric Zoning update action.

The format of the RAPZ Request CT_IU is shown in table 412.

Table 412 – RAPZ Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Name	see 6.4.8.1

The format of the Accept CT_IU to a RAPZ request is shown in table 413.

Table 413 – Accept CT_IU to RAPZ Request

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.4.10.4.33 Get Alias Names (GAN) Operation

The Fabric Zone Server shall, if it receives a GAN operation request, return a list of Alias Names for the specified Zone Member Object defined in the Zone Set Database. The GAN request payload shall specify the Zone Member Object for which the list of alias names are sought. The GAN accept payload shall contain a list of alias names associated with the specified Zone Member Object.

If no Alias Names are currently registered for the specified Zone Member Object, then the number of Alias Name Entries is set to zero in the response. The format of the GAN Request payload is depicted in table 414

Table 414 – GAN Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Zone Member Object	see 6.4.8.3.6

Zone Member Object: The format of the Zone Member Object is described in 6.4.8.3.6. All Zone Member Identifier Types may be used, with the exception of the Alias Name member (type ‘05’).

The format of the GAN Accept payload is depicted in table 415.

Table 415 – GAN Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Number of Alias Name Entries (n) for Aliases with one member	4
Alias Name #1	see 6.4.8.1
Alias Name #2	see 6.4.8.1
...	
Alias Name #n	see 6.4.8.1
Number of Alias Name Entries (m) for Aliases with more than one member	4
Alias Name #1	see 6.4.8.1
Alias Name #2	see 6.4.8.1
...	
Alias Name #m	see 6.4.8.1

The format of the Alias Name is described in 6.4.8.1.

6.5 Security Policy Server

6.5.1 Overview

The Security Policy Server specifies an in-band protocol for controlling and extracting security policy information.

The Security Policy Server is defined to be a Service accessible from the WKA of the Management Service at FFFFFAh. Client requests to the Security Policy Server are addressed to the Management Service subtype shown in table 125.

Security Policy Server requests are carried over the GS Common Transport (FC-CT).

6.5.2 Protocol

Security Policy Server requests are managed through protocols containing a set of Request CT_IUs and Response CT_IUs supported by the Security Policy Server.

For a Security Policy Server request, the payload shall be transported from the requestor to the Security Policy Server using a Request CT_IU. The corresponding Security Policy Server response is transported from the Security Policy Server.

The effect of Server Sessions on the action of the Security Policy Server is specified in FC-SP-2.

For a complete command list and additional details, see FC-SP-2.

6.6 Security Information Server

6.6.1 Overview

The Security Information Server provides an in-band protocol for observing and controlling operational security information. It does not provide any access to Fabric security policy information, which is the object of the Security Policy Server (see 6.5).

Each Security Information Server command may specify restrictions on the use and behavior of the command. Different commands may specify and/or enforce different restrictions.

6.6.2 Protocol

Security Information Server commands are conducted through a set of Request CT_IUs and Response CT_IUs carried over the GS Common Transport (FC-CT).

In the Basic CT_IU preamble of CT_IUs sent to or from the Security Information Server:

a) the GS_Type field shall be set to the WKA of the Management Service (see table 4);

b) the GS_Subtype field shall be set to the Security Information Server subtype (see table 125); and

c) the Command/Response code field shall be set to a defined Security Information Server Command Code (see table 416).

Table 416 – Security Information Server Command Codes

Code	Mnemonic	Description	Constrained by Zoning	Reference subclause
0001h	GAS_ID	Get Authentication State for Port Identifier	yes	6.6.5.1
8001h	Reject	Security Information Server rejected a request	yes	4.4.4
8002h	Accept	Security Information Server accepted a request	yes	6.6.5.1
other	reserved			

The action of the Security Information Server is unaffected by Server Sessions.

6.6.3 Security Information Server Objects and Attributes

6.6.3.1 Overview

All Security Information Server objects are data structures that shall be formatted as shown in table 417.

Table 417 – Security Information Server object format

Item	Size (Bytes)
Object Type	2
Structure Length (L)	2
Object Value	L-4

Object Type shall identify the format and meaning of the Object Value field. The value of the Object Type field shall be one of the values in table 418.

Table 418 – Security Information Server Object Type codes

Code	Description	Reference subclause
0001h	Port Identifier	6.6.3.2
0002h	Authentication State	6.6.3.3
all other	reserved	

Structure Length shall be the length in bytes of the entire Security Information Server object data structure (i.e., the length in bytes of the Object Value field plus four).

Object Value shall be the value of the object. Its size and format are dependent on the Object Type. Its length in bytes shall be a multiple of four.

6.6.3.2 Port Identifier Object Value Format

If the Object Type field in a Security Information Server object is set to Port Identifier (see table 418), the Object Value field shall be set to a Port Identifier Object Value structure (see table 419). The Port Identifier Object Value is a data structure that contains an N_Port_ID.

Table 419 – Port Identifier Object Value format

Item	Size (Bytes)
reserved	1
N_Port_ID (see FC-FS-4)	3

6.6.3.3 Authentication State Object Value Format

If the Object Type field in a Security Information Server object is set to Authentication State (see table 418), the Object Value field shall be set to an Authentication State Object Value structure (see table 420). The Authentication State Object Value is a data structure that indicates whether an Nx_Port is currently authenticated with the Fabric (see FC-SP-2), and may also provide the negotiated Authentication Protocol and Authentication Protocol Parameters by which authentication was completed. A Security Information Server shall not send a response containing an Authentication State object for an Nx_Port that is not currently logged in with the Fabric.

Table 420 – Authentication State Object Value format

Item	Size (Bytes)
State	1
N_Port_ID (see FC-FS-4)	3
Authentication Protocol Identifier	4
Authentication Protocol Parameters	m*4

The State field shall indicate the authentication state with the Fabric (see FC-SP-2) of the N_Port_ID to which it refers, as specified in table 421.

Table 421 – Meaning of the values of the State field

State Value	Meaning
0	The Fx_Port to which the Nx_Port is attached does not support authentication of an Nx_Port with the Fabric
1	The Fx_Port to which the Nx_Port is attached supports authentication of an Nx_Port with the Fabric and the Nx_Port is logged in without authentication.
2	The Fx_Port to which the Nx_Port is attached supports authentication of an Nx_Port with the Fabric and the Nx_Port is logged in with authentication.
All other values	reserved

The N_Port_ID field is the N_Port_ID of the Nx_Port to which the Authentication State Object Value refers.

The Authentication Protocol Identifier field in the Authentication State Object Value shall be set to zero if:

a) the Fx_Port to which the Nx_Port is attached does not support authentication of an Nx_Port with the Fabric; or

b) The Security Information Server does not return Authentication Protocol Identifiers.

The Authentication Protocol Identifier field in the Authentication State Object Value shall be set to the Authentication Protocol Identifier (see FC-SP-2) for the Authentication Protocol by which the referenced Nx_Port is authenticated with the Fabric if:

a) the Fx_Port to which the Nx_Port is attached supports authentication of an Nx_Port with the Fabric; and

b) The Security Information Server returns Authentication Protocol Identifiers.

The Authentication Protocol Parameters field shall have zero length if:

a) The Authentication Protocol Identifier value is zero;

b) The Security Information Server does not return Authentication Protocol Parameters; or

c) the selected Authentication Protocol has no parameters.

The Authentication Protocol Parameters field in the Authentication State Object Value shall be set to the Authentication Protocol Parameters (see FC-SP-2) for the Authentication Protocol indicated by the Authentication Protocol Identifier value if:

a) The Authentication Protocol Identifier value is nonzero;

b) The Security Information Server returns Authentication Protocol Parameters; and

c) The selected Authentication Protocol has parameters.

6.6.4 Reason Code Explanations

Reason Code Explanation values with special meaning in Reject CT_IUs sent by Security Information Servers are shown in table 422.

Table 422 – Security Information Server Reason Code Explanations

Value	Description
01h	Subject Nx_Port is not accessible (e.g., not logged in or not in a common Zone)

6.6.5 Commands

6.6.5.1 Query - Get Authentication State - Port Identifier (GAS_ID)

A GAS_ID request is sent by an Nx_Port (i.e., the Requesting Nx_Port) to determine the authentication state of another Nx_Port (i.e., the Subject Nx_Port) with the Fabric. If a Security Information Server accepts a GAS_ID request, it shall return an Accept CT_IU including an

Authentication State object (see table 417) containing an Authentication State Value for the Subject Nx_Port. The format of the GAS_ID Request CT_IU is depicted in table 423.

Table 423 – GAS_ID Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Port Identifier object for Subject Nx_Port	8

A Security Information Server that supports GAS_ID and receives a properly formatted GAS_ID request shall conduct the following steps before any further validation or response is conducted:

1) Determine if:

    I) the Subject Nx_Port is logged in with the Fabric; and

    II) Zoning (see 6.4) is in effect for the Fabric and the Requesting Nx_Port is a member of any Zone of which the Subject Nx_Port is also a member;

2) If :

    I) Zoning (see 6.4) is in effect for the Fabric and the Requesting Nx_Port is not a member of any Zone of which the Subject Nx_Port is also a member; or

    II) the Subject Nx_Port is not logged in with the Fabric;

    a Security Information Server shall respond to a GAS_ID request with a reject CT_IU containing reason code set to ‘Unable to perform command request’ (see table 10) and reason code explanation set to ‘Subject Nx_Port is not accessible’ (see table 422);

    and

3) If:

    I) the Requesting Nx_Port is not authenticated with the Fabric; or

    II) the request is rejected based on authorization constraints that are outside the scope of this standard;

    a Security Information Server shall respond to a GAS_ID request with a reject CT_IU containing reason code set to ‘Unable to perform command request’ (see table 10) and reason code explanation set to ‘Authorization Exception’ (see table 11).

The format of the GAS_ID Accept CT_IU is depicted in table 424.

Table 424 – Accept CT_IU for GAS_ID Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Authentication State object for Subject Nx_Port	L

The return of Authentication Protocol Identifiers and/or Authentication Protocol Parameters in an Accept CT_IU for a GAS_ID request is optional.

6.7 Fabric Device Management Interface

6.7.1 Overview

The Fabric-Device Management Interface (FDMI) enables the management of devices such as HBAs through the Fabric. The FDMI complements the mechanisms already provided by existing Name Server and Management Server functions.

The FDMI defines a CT based mechanism that provides the standardized Management Server command interface to obtain device information from the Fabric. The FDMI also provides a CT based mechanism that allows end-devices to register certain information with the Fabric. Mechanisms other than CT may be used to obtain device information from the Fabric, but their definition and use are beyond the scope of this standard.

In this standard, the FDMI addresses only HBA type devices. The HBA Management Information defined by FDMI is based on information defined by the HBA API specification (see FC-MI-3 and SM-HBA-2).

6.7.2 FDMI Relationship to the Name Server

The FDMI supplements the services provided by the Name Server. Attributes returned by an FDMI server may or may not include attributes specified for a Name Server. To obtain complete information as provided by the HBA API, the Client may need to obtain device information from both the Name Server and HBA Server

6.7.3 GS_Subtypes

To support Fabric-Device Management, a range of GS_Subtype values on the Management Server are designated for the Fabric-Device Management Interface. This range is defined in table 425.

Table 425 – Fabric-Device Management Interface GS_Subtypes

Encoded value	Description
10h	HBA Management Server
11h to 1Fh	Reserved for future use
others	see table 125

6.7.4 HBA Management Server

6.7.4.1 Overview

The HBA Management Server is defined to allow the registration and retrieval of HBA Management information.

6.7.4.2 Platform Model

The support of HBA management through the Fabric is based on the Platform model shown in figure 21. Specifically, a Platform contains one or more nodes and each node contains one or more Nx_Ports. An HBA may be associated with more than one node, and a node may be associated with multiple HBAs. An Nx_Port shall not be associated with more than one HBA and shall not be associated with more than one node.

Additional N_Port_IDs may be registered for the Physical N_Ports on an HBA. If Additional N_Port_IDs are registered, they shall be subject to all the conditions this clause places on Nx_Ports.

Each HBA is uniquely identified by an HBA Identifier. This allows each HBA to be identified in an HBA Management request using the corresponding HBA Identifier.

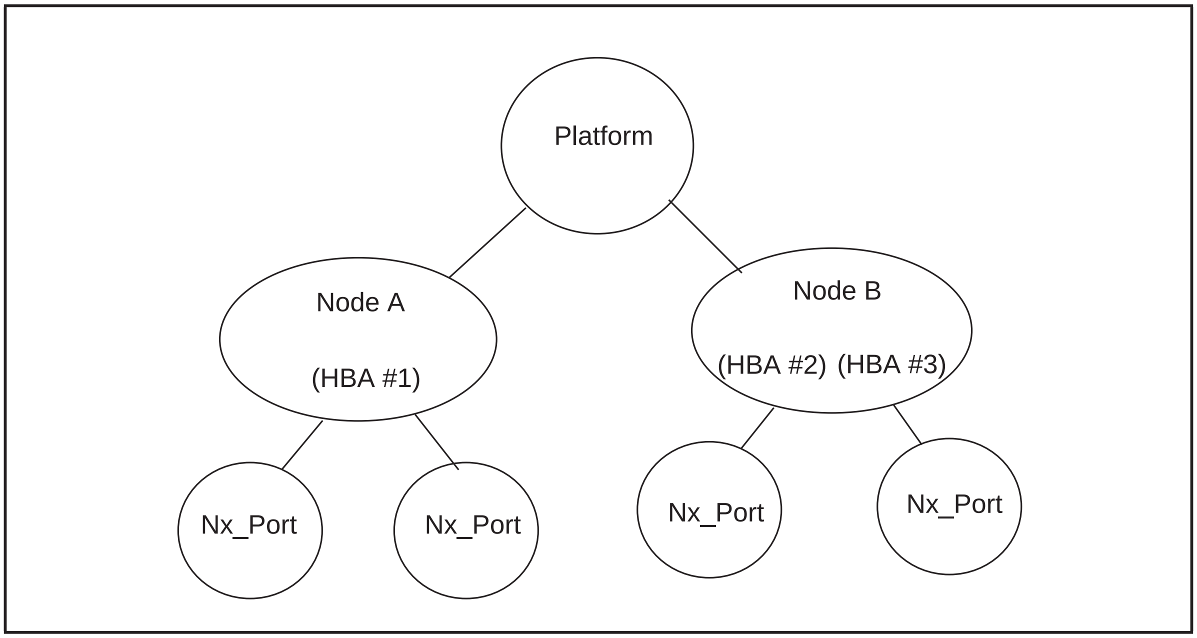


Figure 21 – HBA Management Server Platform Model

6.7.4.3 Protocol

6.7.4.3.1 Overview

HBA Management Server registration, deregistration and queries are managed through protocols containing a set of Request CT_IUs and Response CT_IUs supported by the HBA Management Server.

For a HBA Management Server request, the payload shall be transported from the requestor to the HBA Management Server using a Request CT_IU. The corresponding HBA Management Server response is transported from the HBA Management Server to the requestor, in the Exchange established by the requestor, using a Response CT_IU.

Frames of an HBA Management Server Request CT_IU sent to an HBA Management Server resident in a Fabric shall have their D_ID set to FF FF FAh. Frames of an HBA Management Server Request CT_IU sent to an HBA Management Server resident at an Nx_Port shall have their D_ID set to the N_Port_ID of the Nx_Port.

The action of the HBA Management Server is unaffected by Server Sessions.

6.7.4.3.2 CT_IU Preamble Values

The following values shall be set in the CT_IU preamble for Fabric Configuration Server request and their responses; fields not specified here shall be set as defined in 4.3.2:

a) GS_Subtype: as indicated in table 425; and

b) Command Code: see table 426 for Request command codes.

Table 426 – HBA Management Server - Request Command Codes

Code	Mnemonic	Description	Reference subclause
0100h	GRHL	Get Registered HBA List	6.7.4.7.1
0101h	GHAT	Get HBA Attributes	6.7.4.7.2
0102h	GRPL	Get Registered Port List	6.7.4.7.3
0110h	GPAT	Get Port Attributes	6.7.4.7.4
0120h	GPAS	Get Port Statistics	6.7.4.7.5
0200h	RHBA	Register HBA	6.7.4.7.6
0201h	RHAT	Register HBA Attributes	6.7.4.7.7
0210h	RPRT	Register Port	6.7.4.7.8
0211h	RPA	Register Port Attributes	6.7.4.7.9
0300h	DHBA	Deregister HBA	6.7.4.7.10
0301h	DHAT	Deregister HBA Attributes	6.7.4.7.11
0310h	DPRT	Deregister Port	6.7.4.7.12
0311h	DPA	Deregister Port Attributes	6.7.4.7.13

6.7.4.3.3 Registration

6.7.4.3.3.1 Overview

Registrations are limited to a single HBA Management Server attribute at a time. A registrant submits a tuple, consisting of an object Name_Identifier (see FC-FS-4), along with an attribute to be associated with the object.

The registration requests defined for the HBA Management Server are summarized in table 426. Some attributes do not have a corresponding registration request; this standard does not define the registration of those attributes.

The HBA Management Server may reject registrations due to HBA Management Server resource limitations. However, the HBA Management Server shall support registration of all attributes, once registration of a single attribute has been accepted for a given Name_Identifier.

The HBA Management Server may reject any registration requests for reasons not specified in this standard.

If overlapping registrations for the same attribute are performed, then the HBA Management Server shall, when all registrations have completed, leave the attribute as one of the registered attribute values. However, it is indeterminate which of the overlapping registration requests take precedence.

6.7.4.3.3.2 Registration Rules

The rules for HBA and Nx_Port registration shall be as follows:

a) If a registration or deregistration operation does not originate from an Nx_Port contained in the associated HBA's Registered Port List, then the request is failed with the appropriate reason code explanation;

NOTE 46 – For initial registration of an HBA the Registered Port List is included in the registration request.

b) If a registration for an HBA or Nx_Port is requested and the specified HBA or Nx_Port has been previously registered with the HBA Management Server, then the request is failed with the appropriate reason explanation code. The attributes and their values that were previously registered shall remain unchanged;

c) In order to re-register HBAs or Nx_Ports with the HBA Management Server, the HBA or Nx_Port shall be de-registered prior to a subsequent registration taking place;

d) HBA and Port attributes may be modified without de-registering the Nx_Port or the HBA. In this case attributes and their values that previously existed shall be replaced with the new attribute values. Attributes and their values that did not previously exist are added to the set of attributes. Attributes and their values that exist but are not specified in the new request, remain unchanged;

e) If an Attribute Block contains multiple attributes of the same type, then the request is failed and the appropriate reason code explanation is indicated;

f) If all of the attributes specified in the Attribute Block are not able to be registered with the HBA Management Server, then the request is rejected with the appropriate reason code explanation, and no attributes shall be registered; and

g) Attributes may be implicitly registered with the Name Server and Management Server.

6.7.4.3.4 Queries

The HBA Management Server may reject any query requests for reasons not specified in this standard. The queries defined for the HBA Management Server are summarized in table 426.

6.7.4.4 HBA Management Server Attributes

6.7.4.4.1 Overview

The HBA Management Server includes three types of HBA Management Information:

a) Host Bus Adapter Attributes;

b) Port attributes; and

c) Port Statistics.

Each of these types of HBA Management Information is discussed in this subclause.

6.7.4.4.2 Host Bus Adapter Attributes

6.7.4.4.2.1 Overview

Host Bus Adapter (HBA) attributes are registered with the Name Server or HBA Management Server. HBA attributes may be queried using the appropriate Name Server or HBA Management Server commands.

In the following descriptions, a value may be described as null (i.e. 00h) terminated printable ASCII string. Such values are a concatenation of a number of octets with the value of each octet being equal to or greater than 20h and equal to or less than 7Eh.

NOTE 47 – The maximum length of each attribute is chosen to match the maximum non-null content of the equivalent attribute in the HBA API (see FC-MI-3)

NOTE 48 – Most of the attribute values are intended only to provide information to human administrators. Apart from the constraints stated explicitly here, the values of these attributes are vendor specific. There are no implicit constraints on the values of any attribute (e.g., of uniqueness, format, or consistency with other standards).

HBA attribute descriptions and the means by which they are registered with the Fabric are provided in this subclause.

6.7.4.4.2.2 Manufacturer

The Manufacturer Attribute contains a null (i.e. 00h) terminated printable ASCII string that specifies the manufacturer of the host adapter. The value may match the name by which the manufacturer identifies itself in a telephone directory. This attribute is registered with the HBA Management Server as an HBA attribute. The registration of this attribute is optional.

6.7.4.4.2.3 Serial Number

The Serial Number Attribute contains a null (i.e. 00h) terminated printable ASCII string that specifies the serial number of the host adapter. The value should match a serial number engraved or printed on the host bus adapter, if there is any. The Serial Number Attribute is registered with the HBA Management Server as an HBA attribute. The registration of this attribute is optional.

6.7.4.4.2.4 Model

The Model Attribute contains a null (i.e. 00h) terminated printable ASCII string that specifies the model of the host adapter. The value may match an encoded string used on purchase orders to identify the host adapter model. Some management applications limit this attribute to 63 bytes. The Model Attribute is registered with the HBA Management Server as an HBA attribute. The registration of this attribute is optional.

6.7.4.4.2.5 Model Description

The Model Description Attribute contains a null (i.e. 00h) terminated printable ASCII string that describes the model of the host adapter. The value may provide more detailed or human oriented identification of the model of the host bus adapter than the Model attribute does. The Model

Description Attribute is registered with the HBA Management Server as an HBA attribute. The registration of this attribute is optional.

6.7.4.4.2.6 Node_Name

The Node_Name attribute contains a value that identifies the node that contains the Nx_Ports on the host adapter. If all Nx_Ports on the host bus adapter that are registered with the Name Server are registered the same Node_Name, the Node_Name attribute returned by the HBA Management Server for the host bus adapter shall match the Node_Name attribute registered with the Name Server for its Nx_Ports. If not all Nx_Ports on the host bus adapter that are registered with the Name Server are registered with the same Node_Name, the Node_Name attribute shall not be returned by the HBA Management Server for the host bus adapter.

6.7.4.4.2.7 Node Symbolic Name

The Node Symbolic Name attribute is registered with the Name Server as a Name Server Symbolic Node Name object and is subject to all the description and constraints of that object. The registration of this attribute is optional. If not all Nx_Ports on the host bus adapter that are registered with the Name Server are registered with the same Node_Name, the Node Symbolic Name attribute shall not be returned by the HBA Management Server for the host bus adapter.

6.7.4.4.2.8 Hardware Version

The Hardware Version attribute contains a null (i.e. 00h) terminated printable ASCII string that identifies the hardware version level of the host adapter. Some management applications limit this attribute to 63 bytes. The Hardware Version attribute is registered with the HBA Management Server as an HBA attribute. The registration of this attribute is optional.

6.7.4.4.2.9 Driver Version

The Driver Version attribute contains a null (i.e. 00h) terminated printable ASCII string that identifies the version level of the driver software controlling a host adapter. If a host bus adapter is concurrently under control of multiple driver software modules with different versions, this attribute may indicate the versions for more than one driver module. The Driver Version attribute is registered with the HBA Management Server as an HBA attribute. The registration of this attribute is optional.

6.7.4.4.2.10 Option ROM Version

The Option ROM Version attribute contains a null (i.e. 00h) terminated printable ASCII string that identifies the option ROM or BIOS version of a host adapter. The Option ROM Version attribute is registered with the HBA Management Server as an HBA attribute. The registration of this attribute is optional.

6.7.4.4.2.11 Firmware Version

The Firmware Version attribute contains a null (i.e. 00h) terminated printable ASCII string that identifies the version of firmware executed by a host adapter. If a host bus adapter contains and has the capability to execute multiple firmware modules with different versions, this attribute may indicate the versions for more than one firmware module. The Firmware Version attribute is registered with the HBA Management Server as an HBA attribute. The registration of this attribute is optional.

6.7.4.4.2.12 Operating System Name and Version

The Operating System Name and Version attribute contains a null (i.e. 00h) terminated printable ASCII string that describes the type and version of the operating system controlling the host bus adapter. The Operating System Name and Version attribute is registered with the HBA Management Server as an HBA attribute. The registration of this attribute is optional.

6.7.4.4.2.13 Maximum CT Payload

The Maximum CT Payload attribute is 32-bit unsigned integer equal to the maximum size CT payload in 32-bit words, including all CT headers but no FC frame header(s), that may be sent or received by application software resident in the host in which the host bus adapter is installed. If the host bus adapter does not support generic CT capability for application software on the host in which it is installed, this attribute shall not be registered. The Maximum CT Payload attribute is registered with the HBA Management Server as an HBA attribute. The registration of this attribute is optional.

NOTE 49 – Whether or not an attribute is registered with the HBA Server may depend on the constraints of the resource responsible for the registration. An example of this is where the software responsible for registering HBA attributes may not have access to information associated with some attributes. In this case it is impossible for the software to register that information. This is the main reason that the registration of attributes is optional in most cases.

6.7.4.4.2.14 Vendor Identifier

This attribute shall contain the T10 Vendor ID (see 3.6) of the manufacturer of the HBA, or an OEM of the HBA.

6.7.4.4.2.15 Vendor Specific Information

The Vendor-Specific Information attribute contains a value with vendor specific use. If this attribute is returned for an HBA that supports SM-HBA, it should have the same value as the value returned for the VendorSpecificID HBA attribute of SM-HBA (see SM-HBA-2). The Vendor-Specific ID attribute is registered with the HBA Management Server as an HBA attribute.

6.7.4.4.2.16 Number of Ports

The Number of Ports attribute contains a value equal to the number of Nx_Ports on the HBA (see SM-HBA-2). The Number of Ports attribute is registered with the HBA Management Server as an HBA attribute.

6.7.4.4.2.17 Fabric Name

The Fabric Name attribute contains a value equal to the Fabric_Name of the Fabric associated with the HBA (see FC-SW-6). If the HBA is associated with more than one Fabric, this attribute shall not be registered for the HBA or provided by the HBA. The Fabric Name attribute is registered with the HBA Management Server as an HBA attribute.

6.7.4.4.2.18 Boot BIOS Version

The Boot BIOS Version attribute contains a null (i.e. 00h) terminated printable ASCII string that describes the identification and version of a boot BIOS provided by the HBA. If the HBA does not provide a boot BIOS, this attribute shall not be registered for the HBA or provided by the HBA. The Boot BIOS Version attribute is registered with the HBA Management Server as an HBA attribute. The registration of this attribute is optional.

6.7.4.4.2.19 Boot BIOS State

The Boot BIOS State attribute contains a value that indicates whether a boot BIOS provided by the HBA is enabled or disabled. If the HBA provides a boot BIOS and the boot BIOS is enabled, the value of the Boot BIOS State attribute shall be set to FF FF FF FFh if it is supported. If the HBA provides a boot BIOS and the boot BIOS is disabled, the value of the Boot BIOS State attribute shall be set to zero if it is supported. If the HBA does not provide a boot BIOS, this attribute shall not be registered for the HBA or provided by the HBA. The Boot BIOS State attribute is registered with the HBA Management Server as an HBA attribute. The registration of this attribute is optional.

6.7.4.4.3 Port Attributes

6.7.4.4.3.1 Overview

Port attributes are registered with the Name Server or HBA Management Server. Port attributes may be queried using the appropriate Name Server or HBA Management Server commands.

In the following descriptions, a value may be described as a null (i.e. 00h) terminated printable ASCII string. Such values are a concatenation of a number of octets with the value of each octet being equal to or greater than 20h and equal to or less than 7Eh.

NOTE 50 – The maximum length of each attribute is chosen to match the maximum non-null content of the equivalent attribute in the HBA API (see ~~FC-MI-3~~).

NOTE 51 – Some of the attribute values are intended only to provide information to human administrators. The values of attributes that are registered with the HBA Management Server are subject to no implicit constraints (e.g., of uniqueness, format, or consistency with other standards).

Port attribute descriptions and the means by which they are registered with the Fabric are provided below:

6.7.4.4.3.2 Port Symbolic Name

The Port Symbolic Name attribute is registered with the Name Server as a Name Server Symbolic Port Name object and is subject to all the description and constraints of that object.

6.7.4.4.3.3 Port Identifier

The Port Identifier attribute is registered with the Name Server as a Name Server Port Identifier object and is subject to all the description and constraints of that object.

6.7.4.4.3.4 Port Type

The Port Type attribute is registered with the Name Server as a Name Server Port Type object and is subject to all the description and constraints of that object.

6.7.4.4.3.5 Supported Class of Service

The Supported Class of Service attribute is registered with the Name Server as a Name Server Class of Service object and is subject to all the description and constraints of that object.

6.7.4.4.3.6 Supported FC-4 Types

The Supported FC-4 Types attribute has a format identical to that of a Name Server FC-4 types object. An Nx_Port shall register a Supported FC-4 Types value that indicates “support” for any FC-4 type that it is able to be configured to support. The Supported FC-4 Types attribute is registered with the HBA Management Server as a Port attribute.

6.7.4.4.3.7 Port Active FC-4 Types

The Port Active FC-4 Types attribute is registered with the Name Server as a Name Server FC-4 types object and is subject to all the description and constraints of that object. An Nx_Port shall register a Name Server FC-4 types value that indicates support for any FC-4 type that the Nx_Port is completely configured to support.

6.7.4.4.3.8 Supported Speed

The Supported Speed attribute contains a 32-bit unsigned integer the value of which is a bitmask that indicates in accord with table 427 the Fibre Channel Transmission Speeds that are supported on the PN_Port by which the specified Nx_Port communicates. The Supported Speed attribute is registered with the HBA Management Service as a Port attribute.

Table 427 – Transmission Speed Mask Values

Mask Value	Fibre Channel and Ethernet transmission speed
0000 0001h	1 GFC
0000 0002h	2 GFC
0000 0004h	10 GFC[a]
0000 0008h	4 GFC
0000 0010h	8 GFC
0000 0020h	16 GFC
0000 0040h	32 GFC
0000 0080h	20 GFC

[a] Legacy implementations may have used this bit for Ethernet.

Table 427 – Transmission Speed Mask Values

Mask Value	Fibre Channel and Ethernet transmission speed
0000 0100h	40 GFC[a]
0000 0200h	128 GFC
0000 0400h	64 GFC
0000 0800h	256 GFC
0000 8000h	Speed has not been established or is not able to be determined
0001 0000h	10 GE
0002 0000h	40 GE
0004 0000h	100 GE
0008 0000h	25 GE
0010 0000h	50 GE
0020 0000h	400 GE
other values	Reserved

[a] Legacy implementations may have used this bit for Ethernet.

6.7.4.4.3.9 Current Port Speed

The Current Port Speed attribute contains a 32-bit unsigned integer the value of which is a bitmask that indicates in accord with table 427 the Fibre Channel Transmission Speed at which the PN_Port by which the specified Nx_Port communicates is currently operating. The value of this attribute shall contain only a single mask value from table 427. The Current Port Speed attribute shall be returned by the HBA Management Server as a Port attribute if either it has been registered or it has been determined by the HBA Management Server. If the HBA Management Server is able to determine a speed, the value returned for the Current Port Speed attribute shall indicate the speed determined by the HBA Management Server, regardless of any value registered.

6.7.4.4.3.10 Maximum Frame Size

The Maximum Frame Size attribute contains a 32-bit unsigned integer the value of which is the maximum FC frame payload in bytes. This shall not include the FC header but shall include any optional headers. The Maximum Frame Size attribute is registered with the HBA Management Server as a Port attribute.

6.7.4.4.3.11 OS Device Name

The OS Device Name attribute contains a null (i.e. 00h) terminated printable ASCII string that is recognized as a reference to the Nx_Port by the OS that controls it. If there are several such OS device names that reference the same Nx_Port, this attribute may be a comma-separated list of as many such names as fit in 255 bytes. If the software that registers Nx_Port attributes is unable to determine any such OS Device Name, it shall not register this attribute. The OS Device Name attribute is registered with the HBA Management Server as a Port attribute.

6.7.4.4.3.12 Host Name

The Host Name attribute contains a null (i.e. 00h) terminated printable ASCII string that describes the name of the host associated with the Nx_Port. If there are several such names that reference the same host, this attribute may be a comma-separated list of as many such names as fit in 255 bytes. If the software that registers Nx_Port attributes is unable to determine any such Host Name, it shall not register this attribute. The Host Name attribute is registered with the HBA Management Server as a Port attribute. The registration of this attribute is optional.

6.7.4.4.3.13 Port_Name

The Port_Name attribute is registered with the Name Server and is included in the fixed part of the HBA Management Server Register Port (RPRT) request. It is mandatory to include this attribute in the RPRT request. It ties the HBA Management Information defined in the HBA Management Server with the Nx_Port information registered in the Name Server. If no Nx_Ports are registered for an HBA, information about the HBA that is registered with the Name Server is not available via the HBA Management Server.

6.7.4.4.3.14 Node_Name

The Node_Name attribute contains an 8 byte value that identifies the node that contains the Nx_Port. The format of the Node_Name attribute shall be the format of the Name_Identifier described in FC-FS-4. The Node_Name attribute shall be equal to the ~~Node_Name~~ registered with the Name Server for the Nx_Port. ~~The Node Name attribute is registered with the Name Server.~~

6.7.4.4.3.15 Port Fabric Name

The Port Fabric Name attribute contains an 8 byte binary value equal to the Fabric_Name of the Fabric associated with the Nx_Port (see FC-SW-6). Registration of the Port Fabric Name attribute is optional, and may be registered automatically by the Fabric for Nx_Ports that are registered with the Fabric Name Server. The Port Fabric Name attribute is registered with the HBA Management Server as an Nx_Port attribute.

6.7.4.4.3.16 Port State

The Port State attribute contains the Port State (see SM-HBA-2) for the specified Nx_Port. The Port State is an integer where the value indicates the current state of the Nx_Port. The sequence and timing of Port States that are exhibited due to errors or transient conditions is vendor specific.

6.7.4.4.3.17 Number of Discovered Ports

The Number of Discovered Ports attribute returns the number of FC_Ports that are visible to the Nx_Port identified in the request. At a minimum, this is the number of FC_Ports mapped to a device, but may not reflect all of the nodes on a network.

6.7.4.4.4 Port Statistics

6.7.4.4.4.1 Overview

Port Statistics for an Nx_Port or the PN_Port through which an Nx_Port communicates are obtained from an Nx_Port using CT commands redirected to the Nx_Port by the Fabric. Requests are sent from the management application to the Fabric, and then the Fabric communicates with the specified port to obtain the requested information.

6.7.4.4.4.2 Seconds Since Last Reset

This specifies the number of seconds since the PN_Port or Nx_Port statistics were last reset.

6.7.4.4.4.3 Exchange Count (Originator)

This specifies the total number of Fibre Channel Exchanges opened as Originator by the Nx_Port.

6.7.4.4.4.4 Exchange Count (Responder)

This specifies the total number of Fibre Channel Exchanges opened as Responder by the Nx_Port.

6.7.4.4.4.5 TxSequence Count

This specifies the total number of Fibre Channel Sequences opened as Initiator by the Nx_Port.

6.7.4.4.4.6 RxSequence Count

This specifies the total number of Fibre Channel Sequences opened as Recipient by the Nx_Port.

6.7.4.4.4.7 TxFrame Count

This specifies the total number of Fibre Channel frames transmitted through the Nx_Port.

6.7.4.4.4.8 RxFrame Count

This specifies the total number of Fibre Channel frames received through the Nx_Port.

6.7.4.4.4.9 TxWord Count

This specifies the total number of Fibre Channel words transmitted through the Nx_Port.

6.7.4.4.4.10 RxWord Count

This specifies the total number of Fibre Channel words received through the Nx_Port.

6.7.4.4.4.11 TxKB Count

This specifies the total number of kilobytes transmitted through the Nx_Port.

6.7.4.4.4.12 RxKB Count

This specifies the total number of kilobytes received through the Nx_Port.

6.7.4.4.4.13 LIPCount

This specifies the total number of LIP Primitive Sequences that have occurred on the link attached to the PN_Port (see FC-FS-4).

6.7.4.4.4.14 NOSCount

This specifies the total number of NOS Primitive Sequences that have occurred on the link attached to the PN_Port (see FC-FS-4).

6.7.4.4.4.15 Error Frame Count

This specifies the total number of frames that have been received in error by the PN_Port.

6.7.4.4.4.16 Dumped Frame Count

This specifies the total number of frames that have been discarded by the PN_Port due to the lack of host buffers.

6.7.4.4.4.17 Link Failure Count

This specifies the total number of instances that a link error has occurred for the PN_Port (see FC-FS-4).

6.7.4.4.4.18 Loss of Synchronization Count

This specifies the total number of instances that a loss of synchronization has occurred for the PN_Port (see FC-FS-4).

6.7.4.4.4.19 Loss of Signal Count

This specifies the total number of instances that a loss of signal has occurred for the PN_Port (see FC-FS-4).

6.7.4.4.4.20 Primitive Sequence-Protocol Error Count

This specifies the total number of instances that a Primitive Sequence-Protocol Error has occurred for the PN_Port (see FC-FS-4).

6.7.4.4.4.21 Invalid TxWord Count

This specifies the total number of Invalid Transmission Words received by the PN_Port (see FC-FS-4).

6.7.4.4.4.22 Invalid RxFrame CRC Count

This specifies the total number of frames received by the PN_Port with a CRC error (see FC-FS-4).

6.7.4.4.4.23 RxPBSY Count

This specifies the total number of times the PN_Port has received ~~PBSY~~ in response to a frame (see FC-FS-4).

6.7.4.4.4.24 RxFBSY Count

This specifies the total number of times the PN_Port has received ~~FBSY~~ in response to a frame (see FC-FS-4).

6.7.4.4.4.25 Primitive Sequence Time-out Count

This specifies the total number of times the PN_Port has timed out a Primitive Sequence protocol (see FC-FS-4).

6.7.4.4.4.26 Loop Elastic Buffer Overrun Count

This specifies the total number of times the elastic buffer of the link interface of the PN_Port to an arbitrated loop has overrun (see FC-AL-2).

6.7.4.4.4.27 Loop Arbitration Time-out Count

This specifies the total number of times the PN_Port has timed out arbitration for access to an arbitrated loop (see FC-AL-2).

6.7.4.5 HBA Management Server Objects

6.7.4.5.1 HBA Identifier

An HBA Identifier shall be a 64-bit Name_Identifier that is uniquely associated with the HBA among all HBAs in the same Fibre Channel interaction space (see FC-FS-4). The HBA Identifier for an HBA may be the same as the Name_Identifier of an Nx_Port on the HBA if the required persistence is satisfied. Once an HBA has registered a Name_Identifier as its HBA Identifier, that Name_Identifier shall persist (e.g., across power cycles) as the HBA Identifier for the HBA.

The format of the HBA Identifier is depicted in table 428 below.

Table 428 – HBA Identifier

Item	Size (Bytes)
Name_Identifier	8

6.7.4.5.2 Registered Port List

The format of the Registered Port List is shown in table 429 below.

Table 429 – Registered Port List

Item	Size (Bytes)
Number of Port Entries (n)	4
Port Entry 1	8
Port Entry 2	8
...	
Port Entry n	8

Number of Port Entries: This field specifies the number of Port Entries contained in the Registered Port List. This value shall be greater than or equal to one.

Port Entry: The format of the Port Entry is depicted in table 430 below.

Table 430 – Port Entry

Item	Size (Bytes)
Port_Name	8

6.7.4.5.3 HBA Attribute Block

The HBA Attribute Block is a variable length structure that contains attributes registered for the specified HBA. The format of the HBA Attribute Block is depicted in the table 431 below.

Table 431 – HBA Attribute Block

Item	Size (Bytes)
Number of HBA Attribute Entries (n)	4
HBA Attribute Entry 1	w
HBA Attribute Entry 2	x
...	
HBA Attribute Entry n-1	y
HBA Attribute Entry n	z

Number of HBA Attribute Entries: This field specifies the number of HBA Attribute Entries contained in the HBA Attribute Block. This value shall be greater than or equal to one.

HBA Attribute Entry: An HBA Attribute Entry specifies a particular attribute registered with an HBA object. An HBA Attribute Entry follows the general Attribute Entry format specified in 6.7.4.5.4.

6.7.4.5.4 Attribute Entry Format

The HBA Management Server defines a general format to be used for attributes associated with HBA and Port objects. The general format of the Attribute Entry is depicted in table 432.

Table 432 – Attribute Entry

Item	Size (Bytes)
Attribute Entry Type	2
Attribute Entry Length	2
Attribute Entry Value	see table 433, table 435, and table 437

Attribute Entry Type: This field indicates the Attribute Entry Type. Valid Attribute types are specific to the object to which they are associated. See table 433 for a definition of valid Attribute Entry Types.

Attribute Entry Length: This field indicates the total length of the Attribute Entry. The total length in bytes shall be a multiple of four and includes the Attribute Entry Type, Attribute Entry Length, and Attribute Value fields.

Attribute Entry Value: This field specifies the Attribute Entry Value. Attribute Entry Values shall be at least four bytes in length and the length shall be a multiple of four. For variable length Attribute Value fields, fill bytes are added as necessary to the end of the actual value in order to ensure that the length of the value field is a multiple of four. Fill bytes shall be nulls (00h). The number of fill bytes

(m) is zero, one, two, or three depending on the length of the actual value (n). Therefore the total length of the value field is (n+m). Attribute Entry Values are defined in table 433.

Table 433 – Attribute Entry Types and associated Values

Type	Value			
	Description	**Length (Bytes)**	**Type**	**Required**
0001h	Node_Name	8	Binary	Yes
0002h	Manufacturer	1 to 63	ASCII	No
0003h	Serial Number	1 to 63	ASCII	No
0004h	Model	1 to 255	ASCII	No
0005h	Model Description	1 to 255	ASCII	No
0006h	Hardware Version	1 to 255	ASCII	No
0007h	Driver Version	1 to 255	ASCII	No
0008h	Option ROM Version	1 to 255	ASCII	No
0009h	Firmware Version	1 to 255	ASCII	No
000Ah	OS Name and Version	1 to 255	ASCII	No
000Bh	Maximum CT Payload Length	4	Binary	No
000Ch	Node Symbolic Name	0 to 255	ASCII	No
000Dh	Vendor Specific Information	4	Binary	No
000Eh	Number of Ports	4	Binary	No
000Fh	Fabric Name	8	Binary	No
0010h	Boot BIOS Version	1 to 255	ASCII	No
0011h	Boot BIOS State	4	Boolean	No
00E0h	Vendor Identifier	8	ASCII	No
F000h - FFFFh	Vendor Specific Use			
other values	Reserved			

6.7.4.5.5 Port Attribute Block

The Port Attribute Block is a variable length structure that contains attributes registered for the specified Nx_Port. The format of the Port Attribute Block is depicted in the table 434 below.

Table 434 – Port Attribute Block

Item	Size (Bytes)
Number of Port Attribute Entries (n)	4
Port Attribute Entry 1	w
Port Attribute Entry 2	x
...	
Port Attribute Entry n-1	y
Port Attribute Entry n	z

Number of Port Attributes: This field specifies the number of Port Attribute Entries contained in the Port Attribute Block. A value of zero in this field shall indicate that no attributes have been registered for the specified Nx_Port.

Port Attribute Entry: A Port Attribute Entry specifies a particular attribute registered for a Port object. A Port Attribute Entry follows the general Attribute Entry format specified in 6.7.4.5.4.

6.7.4.5.6 Port Attributes

The valid Port Attributes are depicted in table 435.

Table 435 – Port Attribute Values

Type	Description	Length (Bytes)	Type	Required
0001h	Supported FC-4 Types	32	Bitmask	No
0002h	Supported Speed	4	Bitmask	No
0003h	Current Port Speed	4	Bitmask	No
0004h	Maximum Frame Size	4	Binary	No
0005h	OS Device Name	1 to 255	ASCII	No
0006h	Host Name	1 to 255	ASCII	No
0007h	Node_Name	8	Binary	Yes
0008h	Port_Name	8	Binary	No
0009h	Port Symbolic Name	0 to 255	ASCII	No
000Ah	Port Type	4	Binary	No
000Bh	Supported Classes of Service	4	Bitmask	No
000Ch	Port Fabric Name	8	Binary	No
000Dh	Port Active FC-4 Types	32	Bitmask	No
0101h	Port State	4	Binary	No
0102h	Number of Discovered Ports	4	Binary	No
0103h	Port Identifier	4	Binary	No
F000h - FFFFh	Vendor Specific Use			
other values	Reserved			

6.7.4.5.7 Port Statistics Block

The Port Statistics Block is a variable length structure that returns statistics for the specified Nx_Port. The format of the Port Statistics Block is depicted in the table 436 below.

Table 436 – Port Statistics Block

Item	Size (Bytes)
Number of Port Statistic Entries (n)	4
Port Statistic Entry 1	12
Port Statistic Entry 2	12
...	...
Port Statistic Entry n-1	12
Port Statistic Entry n	12

Number of Port Statistics: This field specifies the number of Port Statistic Entries contained in the Port Statistics Block. A value of zero in this field shall indicate that no attributes are available for the specified Nx_Port.

Port Statistic Entry: A Port Statistic Entry specifies a particular attribute defined for a Port object. A Port Statistic follows the general Attribute Entry format specified in 6.7.4.5.4.

6.7.4.5.8 Port Statistic Values

The valid Port Statistic Values are depicted in table 437.

Table 437 – Port Statistic Values

Type	Description	Length (Bytes)	Type	Required
0001h	Seconds Since Last Reset	8	Binary	No
0002h	Exchange Count (Originator)	8	Binary	No
0003h	Exchange Count (Responder)	8	Binary	No
0004h	TxSequence Count	8	Binary	No
0005h	RxSequence Count	8	Binary	No
0006h	TxFrames Count	8	Binary	No
0007h	RxFrames Count	8	Binary	No
0008h	TxWords Count	8	Binary	No
0009h	RxWords Count	8	Binary	No

Table 437 – Port Statistic Values (Continued)

Type	Description	Length (Bytes)	Type	Required
000Ah	TxKB Count	8	Binary	No
000Bh	RxKB Count	8	Binary	No
000Ch	LIPCount	8	Binary	No
000Dh	NOSCount	8	Binary	No
000Eh	Error Frame Count	8	Binary	No
000Fh	Dumped Frame Count	8	Binary	No
0010h	Link Failure Count	8	Binary	No
0011h	Loss of Synchronization Count	8	Binary	No
0012h	Loss of Signal Count	8	Binary	No
0013h	Primitive Sequence-Protocol Error Count	8	Binary	No
0014h	Invalid TxWord Count	8	Binary	No
0015h	Invalid RxFrame CRC Count	8	Binary	No
0019h	RxPBSY Count	8	Binary	No
001Ah	RxFBSY Count	8	Binary	No
001Bh	Primitive Sequence Time-out	8	Binary	No
001Ch	Loop Elastic Buffer Overrun Count	8	Binary	No
001Dh	Loop Arbitration Time-out Count	8	Binary	No
F000h to FFFFh	Vendor Specific Use	8	Binary	No
other values	Reserved			

6.7.4.6 Reason Code Explanations

A Reject CT_IU (see 4.4.4) shall notify the requestor that the request has been unsuccessfully completed. The first error condition encountered shall be the error reported by the Reject CT_IU.

If a valid HBA Management Server request is not received, the request is rejected with a reason code of “Invalid Command code” and a reason code explanation of “No additional explanation”.

If a HBA Management Server request is rejected with a reason code of ‘Unable to perform command request’, then one of the reason code explanations, shown in table 438, are returned.

Table 438 – Reason Code Explanations

Encoded value	Description
00h	No Additional Explanation
10h	HBA Already Registered
11h	Attributes For Specified HBA Not Registered
12h	HBA Attribute Block Contains Multiple Attributes of the Same Type
13h	Invalid HBA Attribute Block Length
14h	Required HBA Attributes Not Present
15h	Originating Port Not in Registered Port List
16h	HBA Identifier Not in Registered Port List
17h	HBA Not Registered
20h	Port Attributes Not Registered
21h	Port Not Registered
22h	Port Attribute Block Contains Multiple Attributes of the Same Type
23h	Invalid Port Attribute Block Length
24h	Port Already Registered
30h	Port Does Not Support Dynamic Attributes
31h	Port is Not an HBA Port
41h	HBA Does Not Support Dynamic Attributes
others	Reserved

6.7.4.7 Commands

6.7.4.7.1 Query - Get Registered HBA List (GRHL)

The HBA Management Server shall, if it receives a GRHL request, return a list of HBAs registered with the Fabric. The GRHL request payload shall not specify any request parameters. The GRHL accept payload shall contain a list of registered HBAs. One HBA Identifier is returned for each HBA registered with the HBA Management Server.

If no HBAs are currently registered with the HBA Management Server, then the number of HBA Identifiers in the response shall be set to zero.

The format of the GRHL Request payload is depicted in table 439 below.

Table 439 – GRHL Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3

The format of the GRHL Accept payload is depicted in table 440 below.

Table 440 – GRHL Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Number of HBA Identifiers In List (n)	4
HBA Identifier 1	8
HBA Identifier 2	8
...	
HBA Identifier n	8

6.7.4.7.2 Query - Get HBA Attributes (GHAT)

The HBA Management Server shall, if it receives a GHAT request, return the Registered Port List and the HBA Attribute Block for the specified HBA. The GHAT request payload shall specify the HBA Identifier that identifies the HBA for which attributes are sought. The GHAT accept payload contains the Registered Port List and HBA Attribute Block for the HBA specified in the request.

If the specified HBA has not been registered with the HBA Management Server, then the request is rejected with the appropriate reason code explanation.

The format of the GHAT Request payload is depicted in table 441 below.

Table 441 – GHAT Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
HBA Identifier	8

The format of the GHAT Accept payload is depicted in table 442 below.

Table 442 – GHAT Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Registered Port List	see table 429
HBA Attribute Block	see table 431

6.7.4.7.3 Query - Get Registered Port List (GRPL)

The HBA Management Server shall, if it receives a GRPL request, return a list of registered Nx_Ports for the specified HBA. The GRPL request payload shall specify the HBA Identifier for which the registered Nx_Ports are sought. The GRPL accept payload shall contain the Registered Port List.

The format of the GRPL Request payload is depicted in table 443 below.

Table 443 – GRPL Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
HBA Identifier	8

The format of the GRPL Accept payload is depicted in table 444 below.

Table 444 – GRPL Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Registered Port List	see table 429

Registered Port List: The Registered Port List is described in 6.7.4.5.2.

6.7.4.7.4 Query - Get Port Attributes (GPAT)

The HBA Management Server shall, if it receives a GPAT request, return the Port Attribute Block for the specified Nx_Port. The GPAT request payload shall specify the name identifying the Nx_Port for which attributes are sought. The GPAT accept payload contains the Port Attribute Block for the Nx_Port specified in the request.

If the specified Nx_Port has not been registered with the HBA Management Server, then the request is rejected with the appropriate reason code explanation.

The format of the GPAT Request payload is depicted in table 445 below.

Table 445 – GPAT Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Port_Name	8

The format of the GPAT Accept payload is depicted in table 446 below.

Table 446 – GPAT Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Port Attribute Block	see table 434

Port_Name: This field specifies the name of the Nx_Port specified in the GPAT Request.

Port Attribute Block: The format of the Port Attribute Block is described in 6.7.4.5.5.

6.7.4.7.5 Query - Get Port Statistics (GPAS)

The HBA Management Server shall, if it receives a GPAS operation request, return the Port Statistics Block for the specified Nx_Port. The GPAS request payload shall specify the name identifying the Nx_Port for which statistics are sought. The GPAS accept payload contains the Port Statistics block for the Nx_Port specified in the request.

Port Statistics for an Nx_Port or the PN_Port through which an Nx_Port communicates are obtained from an Nx_Port using CT commands redirected to the Nx_Port by the Fabric.

If the specified Nx_Port has not been registered with the HBA Management Server, then the request is rejected with the appropriate reason code explanation.

The format of the GPAS Request payload is depicted in table 447 below.

Table 447 – GPAS Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Port_Name	8

The format of the GPAS Accept payload is depicted in table 448 below.

Table 448 – GPAS Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Port Statistics Block	4 to n

Port_Name: This field specifies the name of the Nx_Port specified in the GPAS Request.

Port Statistics Block: The format of the Port Statistics Block is described in 6.7.4.5.7.

6.7.4.7.6 Register HBA (RHBA)

The HBA Management Server shall if it receives an RHBA request, register the HBA, its Nx_Ports, and HBA attributes for the specified HBA. The RHBA request payload shall specify the HBA Identifier, the Registered Port List and the HBA Attribute Block.

The RHBA request shall originate from an Nx_Port that is contained in the Registered Port List for the designated HBA. In this case the Registered Port List is contained in the RHBA request. If the RHBA request does not originate from an Nx_Port in the Registered Port List for the HBA, then the request is failed with the appropriate reason code explanation.

If the specified attributes are unable to be registered for the HBA, then the HBA, its Nx_Ports, and its attributes are not registered and the request is failed with the appropriate reason code explanation.

Following a successful HBA registration, the HBA Management Server maintains a Registered Port List associated with each HBA for use in subsequent operations.

The format of the RHBA Request payload is depicted in table 449 below.

Table 449 – RHBA Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
HBA Identifier	8
Registered Port List	see table 429
HBA Attribute Block	see table 431

HBA Identifier: The format of the HBA Identifier is described in 6.7.4.5.1.

Registered Port List: The format of the Registered Port List is described in 6.7.4.5.2.

HBA Attribute Block: The format of the HBA Attribute block is described in 6.7.4.5.3.

The format of the RHBA Accept payload is depicted in table 450 below.

Table 450 – RHBA Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.7.4.7.7 Register HBA Attributes (RHAT)

The HBA Management Server shall if it receives an RHAT request, register the HBA attributes for the specified HBA. The RHAT request payload shall specify the HBA Identifier and the HBA Attribute Block.

The format of the RHAT Request payload is depicted in table 451 below.

Table 451 – RHAT Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
HBA Identifier	8
HBA Attribute Block	see table 431

HBA Identifier: The format of the HBA Identifier is described in 6.7.4.5.1.

HBA Attribute Block: The format of the HBA Attribute block is described in 6.7.4.5.3.

The format of the RHAT Accept payload is depicted in table 452 below.

Table 452 – RHAT Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.7.4.7.8 Register Port (RPRT)

The HBA Management Server shall, if it receives an RPRT request, register the specified Nx_Port and its attributes contained in the Port Attribute Block for the specified Nx_Port. In addition, the specified Nx_Port is added to the Registered Port List for the HBA. The RPRT request payload shall specify the HBA Identifier of the associated HBA, the Port_Name, and the Port Attribute Block that contains the attributes to be registered. The RPRT accept payload shall be null.

If the RPRT request does not originate from an Nx_Port in the Registered Port List for the HBA, then the request is failed with the appropriate reason code explanation.

If the specified attributes are unable to be registered for the Nx_Port, then the Nx_Port and its attributes are not registered and the request is failed with the appropriate reason code explanation.

The format of the RPRT Request payload is depicted in table 453 below.

Table 453 – RPRT Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
HBA Identifier	8
Port_Name	8
Port Attribute Block	see table 434

HBA Identifier: The format of the HBA Identifier is described in 6.7.4.5.1.

Port_Name: The name of the Nx_Port for which attributes are to be registered.

Port Attribute Block: The format of the Port Attribute block is described in 6.7.4.5.5.

The format of the RPRT accept payload is depicted in table 454 below.

Table 454 – RPRT Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.7.4.7.9 Register Port Attributes (RPA)

The HBA Management Server shall, if it receives an RPA request, register the attributes contained in the Port Attribute block for the specified Nx_Port. If the specified Nx_Port was not previously registered then the request is failed with the appropriate reason code explanation. The RPA request payload shall specify the Port_Name and the Port Attribute Block that contains the attributes to be registered. The RPA accept payload shall be null.

The format of the RPA Request payload is depicted in table 455 below.

Table 455 – RPA Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Port_Name	8
Port Attribute Block	see table 434

Port_Name: The name of the Nx_Port for which attributes are to be registered.

Port Attribute Block: The format of the Port Attribute block is described in 6.7.4.5.5.

The format of the RPA accept payload is depicted in table 456 below.

Table 456 – RPA Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.7.4.7.10 Deregister HBA (DHBA)

The HBA Management Server shall, if it receives an DHBA request, de-register the HBA and its attributes. In addition, all Nx_Ports registered for the HBA and their attributes shall be de-registered from the HBA Management Server. The DHBA request payload shall specify the HBA Identifier that identifies the HBA to be de-registered.

If the specified HBA has not been registered with the HBA Management Server, then the request is rejected with the appropriate reason code explanation.

The format of the DHBA Request payload is depicted in table 457 below.

Table 457 – DHBA Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
HBA Identifier	8

The format of the DHBA Accept payload is depicted in table 458 below.

Table 458 – DHBA Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.7.4.7.11 Deregister HBA Attributes (DHAT)

The HBA Management Server shall, if it receives an DHAT request, de-register all optional attributes for the specified HBA. The DHAT request payload shall specify the HBA Identifier that identifies the HBA for which its attributes are to be de-registered.

If the specified HBA or its attributes have not been registered with the HBA Management Server, then the request is rejected with the appropriate reason code explanations.

The format of the DHAT Request payload is depicted in table 459 below.

Table 459 – DHAT Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
HBA Identifier	8

The format of the DHBA Accept payload is depicted in table 460 below.

Table 460 – DHAT Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.7.4.7.12 Deregister Port (DPRT)

The HBA Management Server shall, if it receives an DPRT request, de-register the specified Nx_Port and its attributes. The specified Port_Name shall be removed from the Registered Port List. The DPRT request payload shall specify the Port_Name that identifies the Nx_Port to be de-registered. The DPRT accept payload shall be null.

If the specified Nx_Port is the last registered Nx_Port for the HBA, then the HBA and its attributes are also de-registered.

If the specified Nx_Port has not been registered with the HBA Management Server, then the request is rejected with the appropriate reason code explanation.

The format of the DPRT Request payload is depicted in table 461 below.

Table 461 – DPRT Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Port_Name	8

The format of the DPRT accept payload is depicted in table 462 below.

Table 462 – DPRT Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.7.4.7.13 Deregister Port Attributes (DPA)

The HBA Management Server shall, if it receives an DPA request, de-register all optional attributes for the specified Nx_Port. The DPA request payload shall specify the Port_Name that identifies the Nx_Port for which its attributes are to be de-registered.

If attributes for the Nx_Port have not been registered with the HBA Management Server, then the request is rejected with the appropriate reason code explanations.

If the DPA originates from a Nx_Port that is not registered with the same HBA as the Nx_Port to be deregistered, then the request is failed with the appropriate reason code explanation.

The format of the DPA Request payload is depicted in table 463 below.

Table 463 – DPA Request Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3
Port_Name	8

The format of the DPA Accept payload is depicted in table 464 below.

Table 464 – DPA Accept Payload

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.8 Enhanced Fabric Configuration Server

6.8.1 Overview

The Enhanced Fabric Configuration Server provides a way for a management application to discover Fibre Channel Fabric topology and attributes.

Requests for the Enhanced Fabric Configuration Server are carried over the Common Transport (see clause 4).

The Enhanced Fabric Configuration Server is intended to be distributed among Switches, making the Enhanced Fabric Configuration Server immediately available to an Nx_Port once it has successfully completed Fabric Login. However, the Enhanced Fabric Configuration Server is not restricted or required to be part of a Fabric, and may be located in any Nx_Port.

6.8.2 Protocol

6.8.2.1 Overview

Enhanced Fabric Configuration Server registration, deregistration, and queries are managed through protocols containing a set of Request CT_IUs and Response CT_IUs supported by the Enhanced Fabric Configuration Server.

For an Enhanced Fabric Configuration Server request, the payload shall be transported from the requestor to the Enhanced Fabric Configuration Server using a Request CT_IU. The corresponding Enhanced Fabric Configuration Server response is transported from the Enhanced Fabric Configuration Server to the requestor, in the Exchange established by the requestor, using a Response CT_IU.

The action of the Enhanced Fabric Configuration Server is unaffected by Server Sessions.

6.8.2.2 CT_IU preamble values

The following values shall be set in the CT_IU preamble for Enhanced Fabric Configuration Server request and their responses; fields not specified here shall be set as defined in 4.3.2:

a) GS_Subtype: as indicated in table 125; and

b) Command Code: see table 465 for Request command codes.

Table 465 – Enhanced Fabric Configuration Server - Request Command Codes

Code	Mnemonic	Description	Reference subclause
0100h	eGIEL	Get Interconnect Element List	6.8.5.2
0101h	eGIEAB	Get Interconnect Element Attribute Block	6.8.5.3
0102h	eGIEPL	Get Interconnect Element Port List	6.8.5.4
0103h	eGFO	Get Fabric Object	6.8.5.5
0110h	eGPOAB	Get Physical Object Attribute Block	6.8.5.6
0111h	eGPOPL	Get Physical Object Port List	6.8.5.7
0130h	eGLPAB	Get Logical Port Attribute Block	6.8.5.8
0131h	eGAPL	Get Attached Port List	6.8.5.9
0140h	eGPPAB	Get Physical Port Object Attribute Block	6.8.5.10
0200h	eRIELN	Register Interconnect Element Logical Name	6.8.5.11
other	reserved		

6.8.2.3 Registration

The registration requests defined for the Enhanced Fabric Configuration Server are summarized in table 465. This standard does not define the registration of attributes which do not have a corresponding registration request.

The Enhanced Fabric Configuration Server may reject registrations due to Enhanced Fabric Configuration Server resource limitations. However, the Enhanced Fabric Configuration Server shall support registration of all attributes, once registration of a single attribute has been accepted.

The Enhanced Fabric Configuration Server may reject any registration requests for reasons not specified in this document.

If overlapping registrations for the same attribute are performed, then the Enhanced Fabric Configuration Server shall, when all registrations have been completed, leave the attribute as one of the registered attribute values. However, it is indeterminate which of the overlapping registration requests take precedence.

6.8.2.4 Queries

The Enhanced Fabric Configuration Server may reject any query request for reasons not specified in this document. The queries defined for the Enhanced Fabric Configuration Server are summarized in table 465.

6.8.3 Enhanced Fabric Configuration Server Objects and Attributes

6.8.3.1 Overview

Figure 22 illustrates the logical Fabric, consisting of one or more Interconnect Elements, each having one or more Logical Ports. These Logical Ports are connected to other Logical Ports.

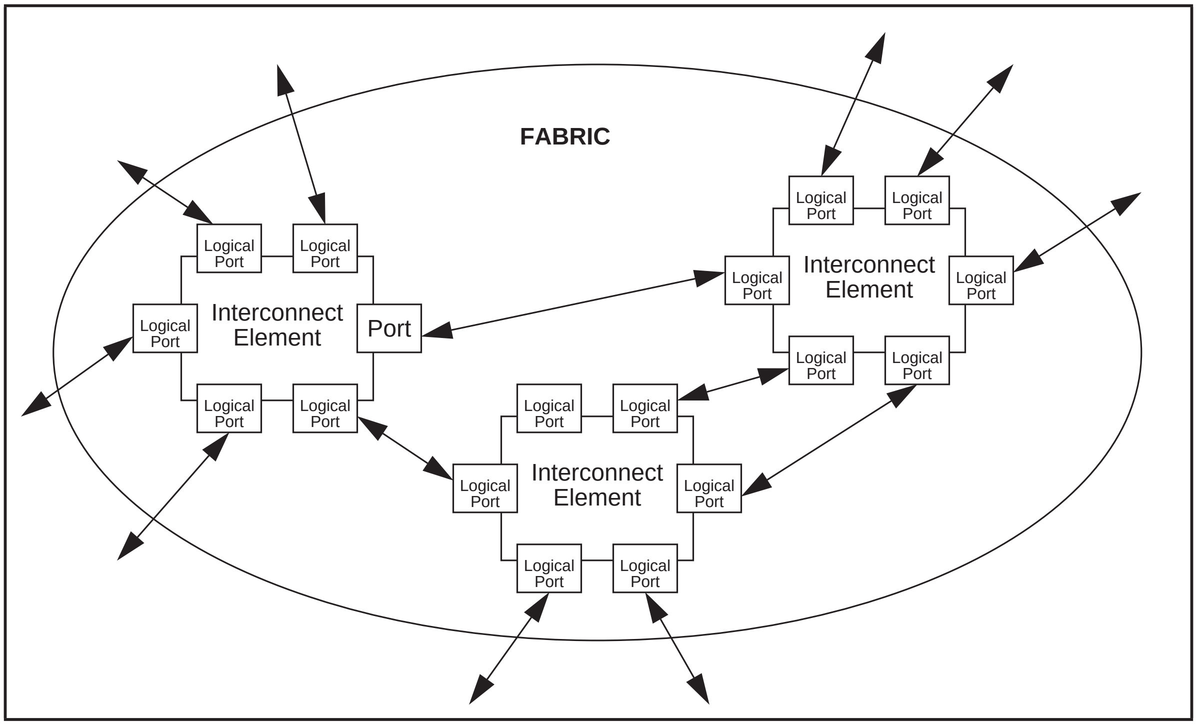


Figure 22 – Logical Fabric Illustration

The Enhanced Fabric Configuration Server object model is shown in figure 23.

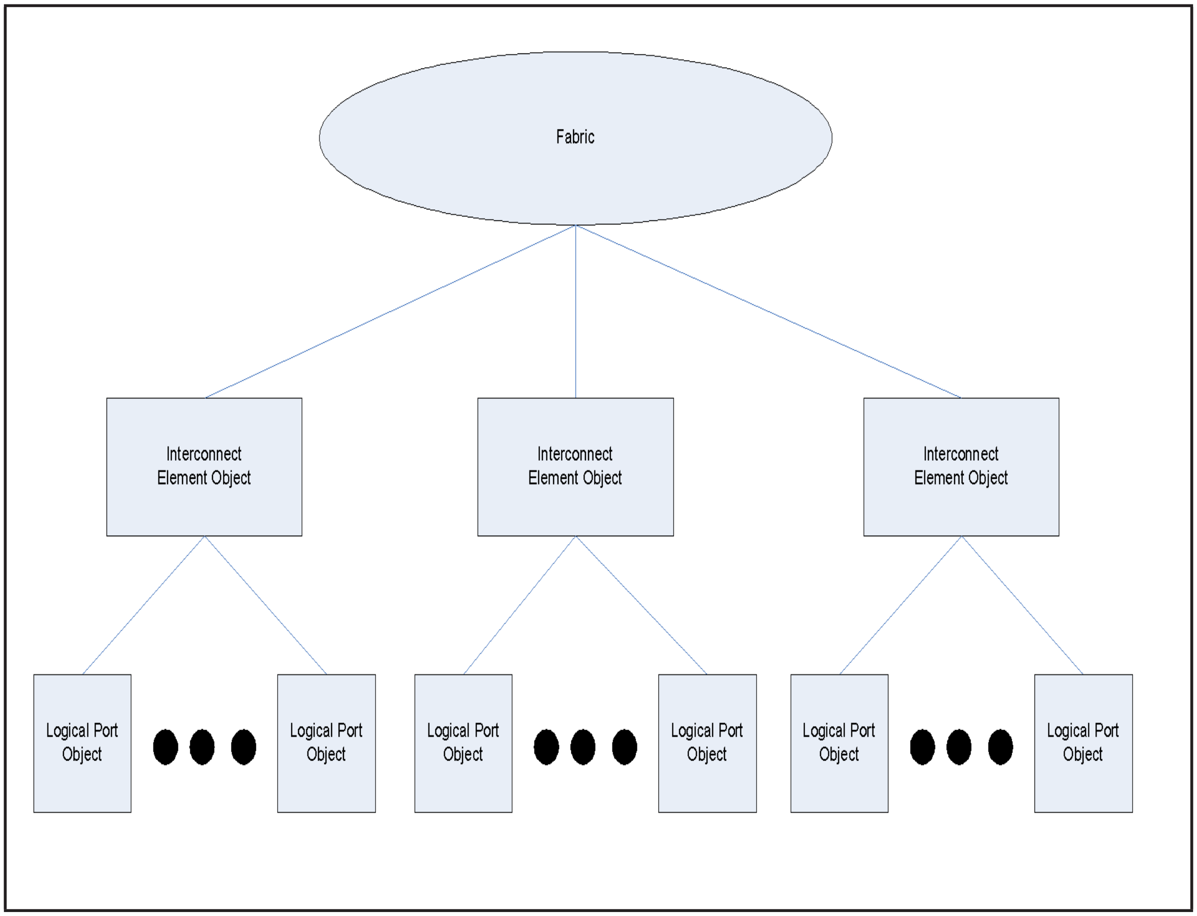


Figure 23 – Enhanced Fabric Configuration Server Logical Model

The base object class managed by the Enhanced Fabric Configuration Server is the Fabric object which has one or more associated Interconnect Element objects. Interconnect Element objects have one or more associated Port objects, and one or more Interconnect Element objects belong to a fabric identified by the object's Fabric Name attribute. One or more Interconnect Element objects may be associated to a Physical object as shown in figure 23. One or more Logical Port objects may be associated to a Physical Port object as shown in figure 24.

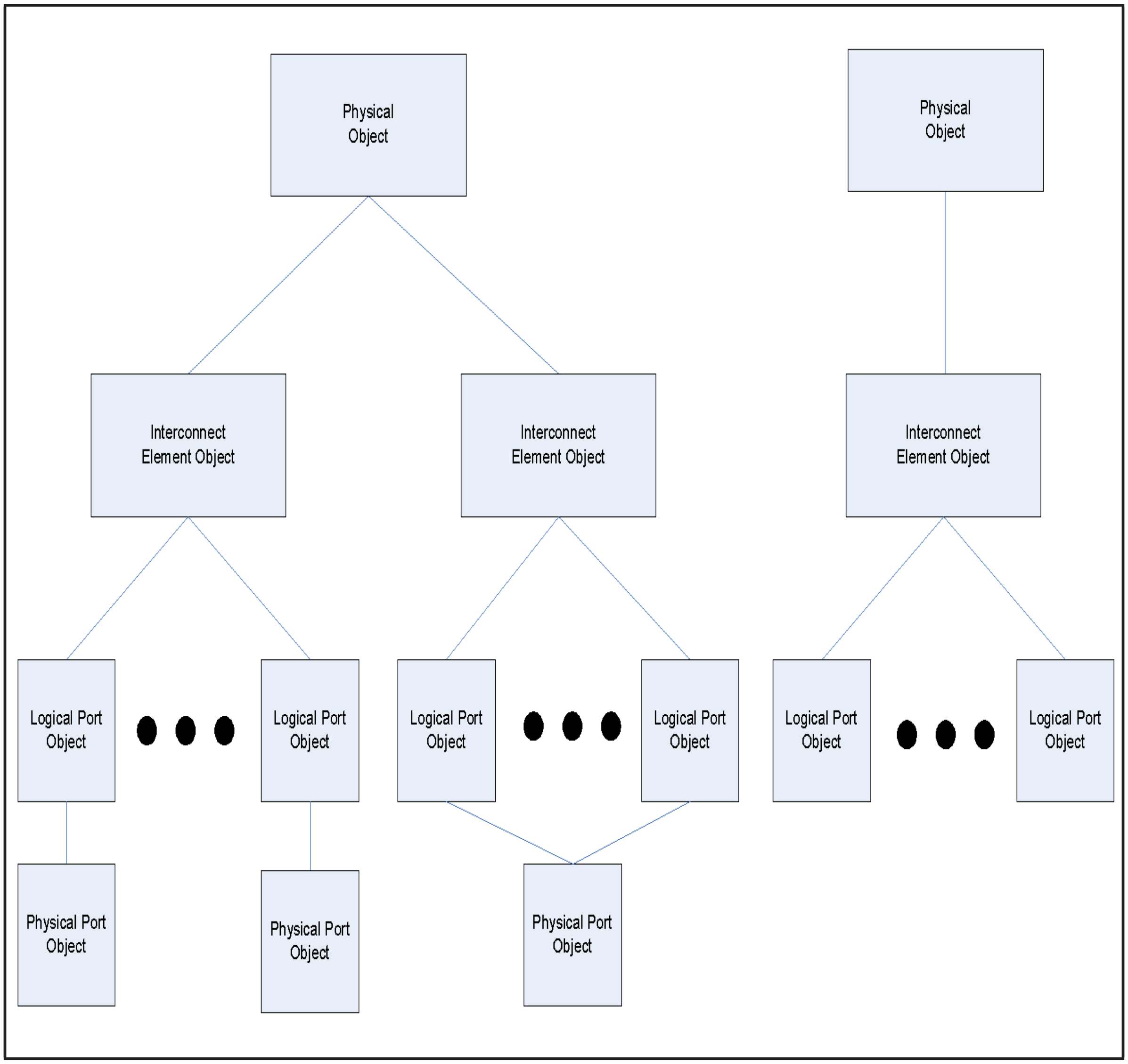


Figure 24 – Enhanced Fabric Configuration Server Logical and Physical Object Model

The Interconnect Element object and Port object may have one or more attributes associated with them, as shown in figure 25. and figure 26.

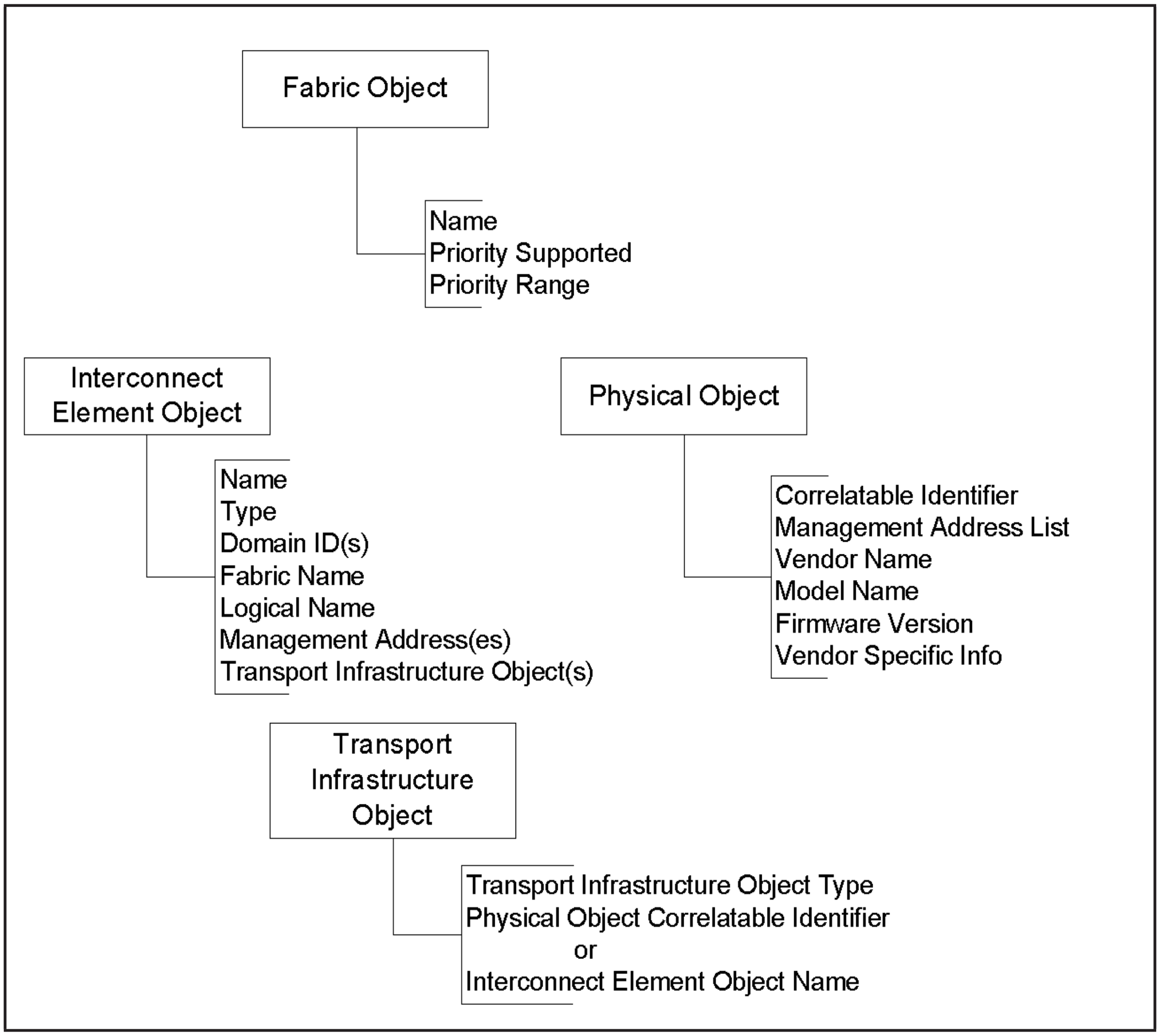


Figure 25 – Enhanced Fabric Configuration Server Logical and Physical Object Attributes

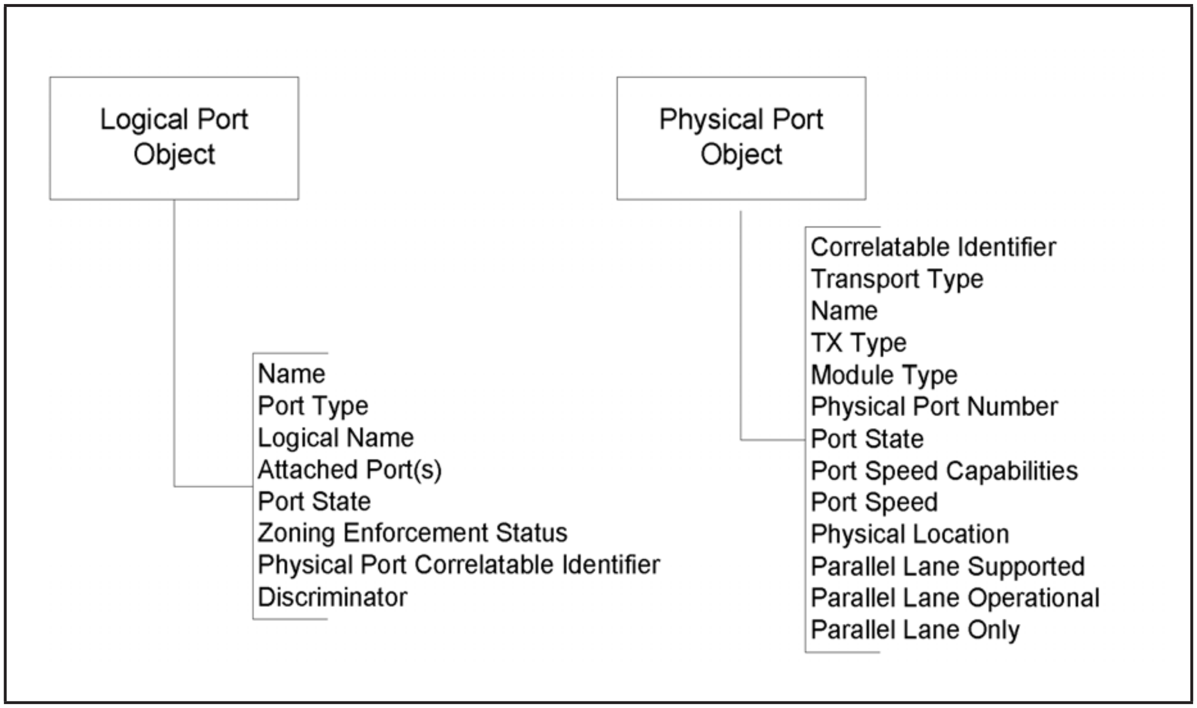


Figure 26 – Logical Port Object and Physical Port Object attributes

6.8.3.2 Enhanced Fabric Configuration Server Attribute Entry Format

6.8.3.2.1 Overview

The Enhanced Fabric Configuration Server defines a general format to be used for attributes associated with Fabric objects. In figure 27, an overview of the relationship of the TLV tables for the Enhanced Fabric Configuration Server is shown. The diagram shows that each object attribute block is comprised of one or many TLV entries. The type of the TLV entry is defined by the object attribute entry types. The general format of the Attribute Entry is depicted in table 466.

Table 466 – Enhanced Fabric Configuration Server Attribute Entry Format

Item	Size (Bytes)
Enhanced Fabric Configuration Server Attribute Entry Type	2
Enhanced Fabric Configuration Server Attribute Entry Length (n)	2
Enhanced Fabric Configuration Server Attribute Entry Value	(see table 468, table 472, table 477, table 479, and table 488)

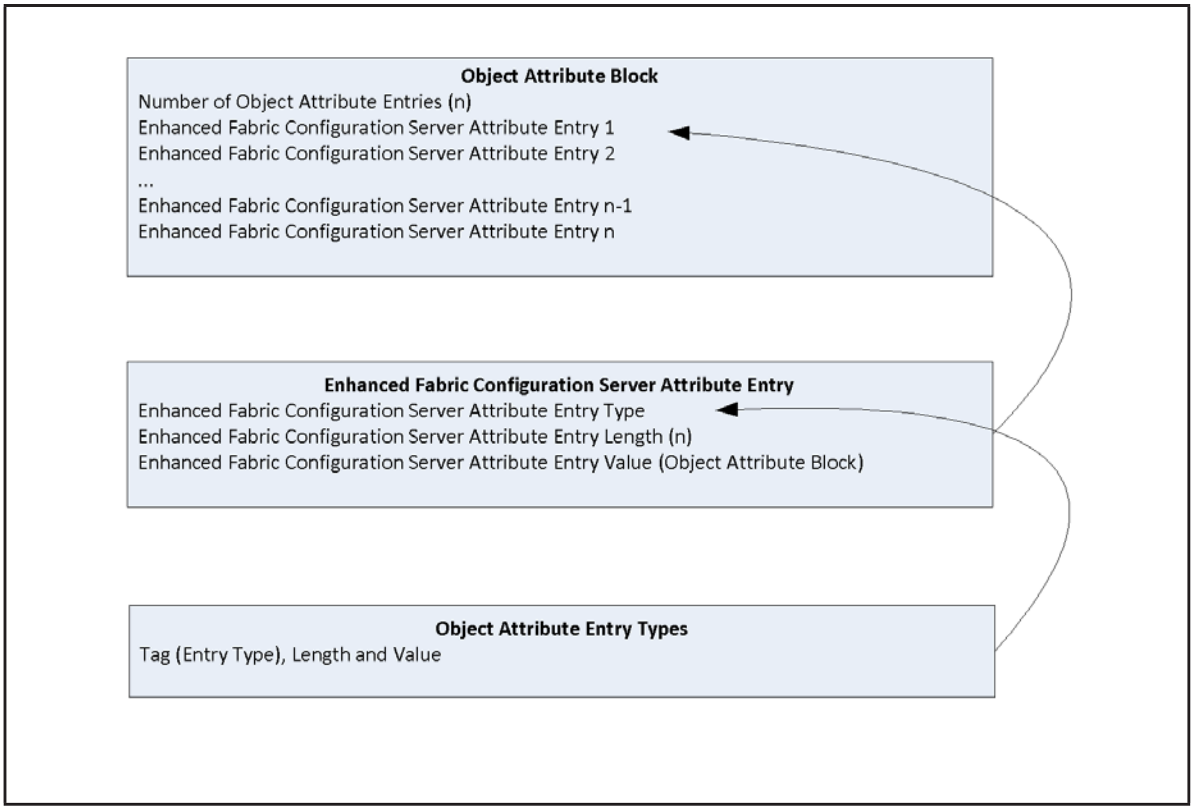


Figure 27 – Overview of the relationship of the TLV tables

6.8.3.2.2 Enhanced Fabric Configuration Server Attribute Entry Type

This field indicates the Attribute Entry Type. Valid Enhanced Fabric Configuration Server Attribute Types are specific to the object to which they are associated. The Type codes are defined in table 472, table 477, table 479, and table 488.

6.8.3.2.3 Enhanced Fabric Configuration Server Attribute Entry Length

This field indicates the total length of the Attribute Entry. The total length in bytes shall be a multiple of four and includes the Attribute Entry Type, Attribute Entry Length, and Attribute Value fields.

6.8.3.2.4 Enhanced Fabric Configuration Server Attribute Entry Value

This field specifies the Enhanced Fabric Configuration Attribute Entry Value. Attribute Entry values are defined in table 472, table 477, table 479, and table 488. Enhanced Fabric Configuration Server Attribute Entry Values shall be at least four bytes in length and the length shall be a multiple of four. For variable length Attribute Value fields, fill bytes are added as necessary to the end of the actual value in order to ensure that the length of the value field is a multiple of four. Fill bytes shall be 00h.

6.8.3.3 Fabric Object

6.8.3.3.1 Fabric Object Attribute Block

6.8.3.3.1.1 Overview

The Fabric Object Attribute Block is a variable length structure that contains attributes registered for the specified Fabric Object. The format of the Fabric Object Attribute Block is depicted in table 467.

Table 467 – Fabric Object Attribute Block

Item	Size (Bytes)
Number of Fabric Object Attribute Entries (n)	4
Enhanced Fabric Configuration Server Attribute Entry 1	see table 468
Enhanced Fabric Configuration Server Attribute Entry 2	see table 468
...	
Enhanced Fabric Configuration Server Attribute Entry n-1	see table 468
Enhanced Fabric Configuration Server Attribute Entry n	see table 468

6.8.3.3.1.2 Number of Fabric Object Attribute Entries

This field specifies the number of Fabric Object Attribute Entries contained in the Fabric Object Attribute Block. This value shall be greater than or equal to one.

6.8.3.3.1.3 Enhanced Fabric Configuration Server Attribute Entry

An Enhanced Fabric Configuration Server Attribute Entry specifies a particular attribute registered with the Fabric Object.

6.8.3.3.2 Fabric Object Attributes

6.8.3.3.2.1 Overview

The Fabric Object Attribute Entry Types and their associated values are depicted in table 468.

Table 468 – Fabric Object Attribute Entry Types and their associated Values

Tag Code	Value				
	Description	**Length (Bytes)**	**Data Type**	**Required**	**Multiples allowed[a]**
0001h	Name	8	Binary	Yes	No
0002h	Priority Supported	1	Binary	Yes	No
0003h	Priority Range	2	Binary	No	Yes
other values	Reserved				

[a] If a Fabric Object Attribute Block contains multiple types for a type that does not allow multiples the command shall be rejected with a reason code of ‘Unable to perform command request’ and a reason code explanation of “Interconnect Element Object Attribute Block Contains Multiple Attributes of the Same Type”.

6.8.3.3.2.2 Fabric Name attribute

The format of the Fabric Name attribute shall be identical to the Name_Identifier format (see FC-FS-4) and the value shall be Fabric_Name (see FC-SW-6).

This standard does not define how this attribute is registered with the Enhanced Fabric Configuration Server.

6.8.3.3.2.3 Priority Supported attribute

If the Priority Supported attribute in the Fabric Object is set to one, at least one Priority Range attribute shall be present in the Fabric Object. If the Priority Supported attribute in the Fabric Object is set to zero, then no Priority Range attributes shall be present in the Fabric Object. The format of the Priority Supported attribute shall be as shown in table 469.

Table 469 – Fabric Priority Supported attribute

Item	Size (Bytes)
Priority Supported	1

This standard does not define how this attribute is registered with the Enhanced Fabric Configuration Server.

6.8.3.3.2.4 Priority Range attribute

The Priority Range attribute in the Fabric Object indicates a range of priority values (see FC-FS-4) having common behavior. Multiple Priority Range attributes may be present in a Fabric Object, each indicating an increasing relative priority. The priority values described by multiple Priority Range attributes shall be contiguous. The format of the Priority Range attribute shall be as shown in table 470.

Table 470 – Fabric Priority Range attribute

Item	Size (Bytes)
Low Range Value	1
High Range Value	1

Low Range Value: contains the lowest value of the described priority range, inclusive.

High Range Value: contains the highest value of the described priority range, inclusive.

This standard does not define how this attribute is registered with the Enhanced Fabric Configuration Server.

6.8.3.4 Interconnect Element Object

6.8.3.4.1 Interconnect Element Object Attribute Block

6.8.3.4.1.1 Overview

The Interconnect Element Object Attribute Block is a variable length structure that contains attributes registered for the specified Interconnect Element Object. The format of the Interconnect Element Object Attribute Block is depicted in table 471.

Table 471 – Interconnect Element Object Attribute Block

Item	Size (Bytes)
Number of Interconnect Element Object Attribute Entries (n)	4
Enhanced Fabric Configuration Server Attribute Entry 1	see table 472
Enhanced Fabric Configuration Server Attribute Entry 2	see table 472
...	
Enhanced Fabric Configuration Server Attribute Entry n-1	see table 472
Enhanced Fabric Configuration Server Attribute Entry n	see table 472

6.8.3.4.1.2 Number of Interconnect Element Object Attribute Entries

This field specifies the number of Interconnect Element Object Attribute Entries contained in the Interconnect Element Object Attribute Block. This value shall be greater than or equal to one.

6.8.3.4.1.3 Enhanced Fabric Configuration Server Attribute Entry

An Enhanced Fabric Configuration Server Attribute Entry specifies a particular attribute registered with an Interconnect Element Object.

6.8.3.4.2 Interconnect Element Object Attributes

6.8.3.4.2.1 Overview

The Interconnect Element Object Attribute Entry Types and their associated values are depicted in table 472.

Table 472 – Interconnect Element Object Attribute Entry Types and their associated Values

Tag Code	Value				
	Description	Length (Bytes)	Data Type	Required	Multiples allowed[a]
0001h	Name	8	Binary	Yes	No
0002h	Type	1	Binary	Yes	No
0003h	Domain ID	1	Binary	Yes[b]	Yes
0004h	Fabric Name	8	Binary	Yes	No
0005h	Logical Name	1 to 255	ASCII	No	No
0006h	Management Address	1 to 255	ASCII	No	Yes
0007h	Transport Infrastructure Object	1 to 255	Binary or ASCII[c]	No	Yes
other values	Reserved				

[a] If an Interconnect Element Object Attribute Block contains multiple types for a type that does not allow multiples, the command shall be rejected with a reason code of ‘Unable to perform command request’ and a reason code explanation of “Interconnect Element Object Attribute Block Contains Multiple Attributes of the Same Type”.

[b] If an Interconnect Element Object Attribute Interconnect Element Type is Switch, then the TLV entry with Interconnect Element Object Attribute Domain ID is required. If the Interconnect Element Object Attribute Interconnect Element Type is not Switch, then there shall be no TLV entry with Interconnect Element Object Attribute Domain ID.

[c] See 6.8.3.4.2.8 .

6.8.3.4.2.2 Interconnect Element Object Name attribute

The format of the Interconnect Element Object Name attribute shall be identical to the Name_Identifier format. If the Interconnect Element is a Switch (see FC-SW-6), the Interconnect Element Object Name attribute shall be the Switch_Name of the Switch.

This standard does not define how this attribute is registered with the Enhanced Fabric Configuration Server.

6.8.3.4.2.3 Interconnect Element Object Type attribute

The values of the Interconnect Element Object Type attribute shall be as shown in table 473.

Table 473 – Interconnect Element Type- encoding

Encoded value	Description
00h	Unknown
01h	Switch
02h	Hub
03h	Bridge
04h	Virtual Domain Supervisor
05h	Controlling Switch
06h	Switch in Principal Domain
07h	FCDF
all others	Reserved

This standard does not define how this attribute is registered with the Enhanced Fabric Configuration Server. The null Interconnect Element Type attribute value is set to ‘Unknown’.

6.8.3.4.2.4 Interconnect Element Object Domain Identifier attribute

The format of the Interconnect Element Object Domain Identifier attribute shall be identical to the Domain Identifier format (see FC-SW-6).

This standard does not define how this attribute is registered with the Enhanced Fabric Configuration Server.

6.8.3.4.2.5 Interconnect Element Object Fabric Name attribute

The format of the Interconnect Element Object Fabric Name attribute shall be identical to the Name_Identifier ~~formata nd~~ the value shall be Fabric_Name (see FC-SW-6).

This standard does not define how this attribute is registered with the Enhanced Fabric Configuration Server. There shall be no null value.

6.8.3.4.2.6 Interconnect Element Object Logical Name attribute

The format of the Interconnect Element Object Logical Name attribute shall be as shown in table 472. The contents of these bytes are not defined and shall not be restricted by the Enhanced Fabric Configuration Server.

This attribute may be registered using the protocol described in 6.8.2.3. The null value for the Interconnect Element Logical Name attribute is a zero-length Interconnect Element Logical Name.

6.8.3.4.2.7 Interconnect Element Object Management Address attribute

The format of the Interconnect Element Object Management Address attribute shall be as shown in table 472. Zero or more Management Address attributes shall be associated with an Interconnect Element object.

The format of the Management Address shall use the format of the Uniform Resource Locator (URL) as defined in RFC3986, RFC4248 and RFC4266. The scheme field shall be as registered at http://www.iana.org/assignments/uri-schemes (see RFC3986). A null management address entry is specified as a Management Address length value of zero followed by 255 reserved bytes.

This standard does not define how this attribute is registered with the Enhanced Fabric Configuration Server. The contents of the Management Address shall not be restricted by the Enhanced Fabric Configuration Server.

6.8.3.4.2.8 Interconnect Element Object Transport Infrastructure Object

The format of the Transport Infrastructure Object shall be shown as in table 474.

Table 474 – Transport Infrastructure Object Format

Item	Size (Bytes)
Transport Infrastructure Object Type	1
Enhanced Fabric Configuration Server Attribute	see table 466
Reserved	2

The Transport Infrastructure Object Type contains either a 1 to indicate the Transport Infrastructure Object references a Physical Object (Physical Object Correlatable Identifier) or a 2 to indicate it references an Interconnect Element Object (Interconnect Element Object Name) as shown in table 475.

Table 475 – Transport Infrastructure Object Attribute Entry Types and their associated Values

Tag	Value				
	Description	Length (Bytes)	Data Type	Required	Multiples allowed[a]
0001h	Physical Object Correlatable Identifier	1 to 255	ASCII	Yes	No
0002h	Interconnect Element Object Name	8	Binary	Yes	No
other values	Reserved				

[a] If a Transport Infrastructure Object Attribute Block contains multiple types for a type that does not allow multiples, then the command shall be rejected with a reason code of ‘Unable to perform command request’ and a reason code explanation of “Transport Infrastructure Object Attribute Block Contains Multiple Attributes of the Same Type”.

6.8.3.5 Physical Object

6.8.3.5.1 Physical Object Attribute Block

6.8.3.5.1.1 Overview

The Physical Object Attribute Block is a variable length structure that contains attributes registered for the specified Physical Object. The format of the Physical Object Attribute Block is depicted in table 476. A Physical Object is the hardware container for an Interconnect Element (e.g. Switch).

Table 476 – Physical Object Attribute Block

Item	Size (Bytes)
Number of Physical Object Attribute Entries (n)	4
Enhanced Fabric Configuration Server Attribute Entry 1	see table 477
Enhanced Fabric Configuration Server Attribute Entry 2	see table 477
...	
Enhanced Fabric Configuration Server Attribute Entry n-1	see table 477
Enhanced Fabric Configuration Server Attribute Entry n	see table 477

6.8.3.5.1.2 Number of Physical Object Attribute Entries

This field specifies the number of Physical Object Attribute Entries contained in the Physical Object Attribute Block. This value shall be greater than or equal to one.

6.8.3.5.1.3 Enhanced Fabric Configuration Server Attribute Entry

An Enhanced Fabric Configuration Server Attribute Entry specifies a particular attribute registered with a Physical Object.

6.8.3.5.2 Physical Object Attributes

6.8.3.5.2.1 Overview

The Physical Object Attribute Entry Types and their associated values are depicted in table 477.

Table 477 – Physical Object Attribute Entry Types and their associated Values

Tag	Value				
	Description	Length (Bytes)	Data Type	Required	Multiples allowed[a]
0001h	Correlatable Identifier	1 to 255	ASCII	Yes	No
0002h	Management Address List	1 to 255	ASCII	No	Yes
0003h	Vendor Name	1 to 255	ASCII	Yes	Yes
0004h	Model Name	1 to 255	ASCII	No	No
0005h	Firmware Version	1 to 255	ASCII	No	Yes
0006h	Vendor Specific Information	1 to 255	ASCII	No	No
other values	Reserved				

[a] If a Physical Object Attribute Block contains multiple types for a type that does not allow multiples, then the command shall be rejected with a reason code of ‘Unable to perform command request’ and a reason code explanation of “Physical Object Attribute Block Contains Multiple Attributes of the Same Type”.

6.8.3.5.2.2 Physical Object Correlatable Identifier attribute

A printable ASCII string, terminated with a null (00h), that uniquely identifies the Physical Object.

This standard does not define how this attribute is registered with the Enhanced Fabric Configuration Server. The contents of the Physical Object Correlatable Identifier shall not be restricted by the Enhanced Fabric Configuration Server.

6.8.3.5.2.3 Physical Object Management Address attribute

A printable ASCII string, terminated with a null (00h), that specifies the model name of the Physical Object.

The format of the Management Address shall use the format of the Uniform Resource Locator (URL) as defined in RFC3986, RFC4248 and RFC4266. The scheme field shall be as registered at http://www.iana.org/assignments/uri-schemes (see RFC3986). A null management address entry is specified as a Management Address length value of zero followed by 255 reserved bytes.

This standard does not define how this attribute is registered with the Enhanced Fabric Configuration Server. The contents of the Management Address shall not be restricted by the Enhanced Fabric Configuration Server.

6.8.3.5.2.4 Physical Object Vendor Name attribute

A printable ASCII character string, terminated with a null (00h), that specifies the vendor name of the Physical Object.

6.8.3.5.2.5 Physical Object Model Name attribute

A printable ASCII character string, terminated with a null (00h), that specifies the user-friendly name of the Physical Object.

6.8.3.5.2.6 Physical Object Firmware Version attribute

A printable ASCII character string, terminated with a null (00h), that specifies the version of the firmware of the Physical Object.

6.8.3.5.2.7 Physical Object Vendor Specific Information attribute

One or more printable ASCII character strings, each terminated with a null (00h), that contain other vendor-specific information regarding the designated Interconnect Element.

6.8.3.6 Logical Port Object

6.8.3.6.1 Logical Port Object Attribute Block

6.8.3.6.1.1 Overview

The Logical Port Object Attribute Block is a variable length structure that contains attributes registered for the specified Logical Port Object. The format of the Logical Port Attribute Block is depicted in table 478.

Table 478 – Logical Port Object Attribute Block

Item	Size (Bytes)
Number of Logical Port Object Attribute Entries (n)	4
Enhanced Fabric Configuration Server Attribute Entry 1	see table 479
Enhanced Fabric Configuration Server Attribute Entry 2	see table 479
...	
Enhanced Fabric Configuration Server Attribute Entry n-1	see table 479
Enhanced Fabric Configuration Server Attribute Entry n	see table 479

6.8.3.6.1.2 Number of Logical Port Object Attribute Entries

This field specifies the number of Logical Port Object Attribute Entries contained in the Logical Port Object Attribute Block. This value shall be greater than or equal to one.

6.8.3.6.1.3 Enhanced Fabric Configuration Server Attribute Entry

An Enhanced Fabric Configuration Server Attribute Entry specifies a particular attribute registered with a Logical Port Object.

6.8.3.6.2 Logical Port Object Attributes

6.8.3.6.2.1 Overview

The Logical Port Object Attribute Entry Types and their associated values are depicted in table 479.

Table 479 – Logical Port Object Attribute Entry Types and their associated Values

Tag	Value				
	Description	Length (Bytes)	Data Type	Required	Multiples allowed[a]
0001h	Name	8	Binary	Yes	No
0002h	Port Type	1	Binary	Yes	No
0003h	Logical Name	1 to 255	ASCII	No	No
0004h	Attached Port	8	Binary	Yes[b]	Yes
0005h	Port State	1	Binary	No	No
0006h	Zoning Enforcement Status	1 to 255	ASCII	No	Yes
0007h	Physical Port Correlatable Identifier	1 to 255	ASCII	No	No
0008h	Discriminator	1	Binary	No	No
other values	Reserved				

[a] If a Logical Port Attribute Block contains multiple types for a type that does not allow multiples the command shall be rejected with a reason code of ‘Unable to perform command request’ and a reason code explanation of “Logical Port Object Attribute Block Contains Multiple Attributes of the Same Type”.

[b] Attached Port is required if Port State is Online.

6.8.3.6.2.2 Logical Port Object Name attribute

The format of the Logical Port Object Name attribute shall be identical to the Name_Identifier format. The value of the Name attribute shall be the same as the value Port_Name in the Fabric Login ELS Accept payload.

This standard does not define how this attribute is registered with the Enhanced Fabric Configuration Server. The null value for the Port Name attribute is 00 00 00 00 00 00 00 00h.

6.8.3.6.2.3 Logical Port Object Port Type attribute

The values of the Port Type attribute shall be as shown in table 480.

Table 480 – Port Type encoding

Encoded value	Description
00h	Unidentified
01h	N_Port
02h	NL_Port
03h	F/NL_Port
7Fh	Nx_Port
81h	F_Port
82h	FL_Port
84h	E_Port
85h	B_Port
86h	A_Port
C0h to FFh	Vendor Specific
all others	Reserved

This standard does not define how this attribute is registered with the Enhanced Fabric Configuration Server. The null Port Type attribute value is set to ‘Unknown’.

6.8.3.6.2.4 Logical Port Object Logical Name attribute

The format of the Logical Port Object Logical Name attribute shall be as shown in table 479. The contents of these bytes are not defined and shall not be restricted by the Enhanced Fabric Configuration Server.

The null value for the Logical Port Object Logical Name attribute is a zero-length Logical Port Logical Name.

6.8.3.6.2.5 Logical Port Object Attached Port Name attribute

The format of the Attached Port Name attribute shall be as shown in table 481. Zero or more Attached Port Name attributes may be associated with a Port object.

Table 481 – Attached Port Name Format

Item	Size (Bytes)
Port Name	8
Reserved	2
Port Flags	1
Port Type	1

Port Name: As defined in 6.2.3.3.1.

Port Flags: As shown in table 482.

Table 482 – Port Flags field bits

Bit Position	Description
7 to 2	Reserved
1	A value of one indicates that the Port supports the Get Topology Information Extended (GTIN) Link Service. A value of zero indicates that the Port does not support this ELS.
0	Obsolete

Port Type: As shown in table 480.

This standard does not define how this attribute is registered with the Enhanced Fabric Configuration Server. A Port object with a Port Type attribute value of “N_Port” or “NL_Port” shall have a null Attached Port Name List. The null value for the Attached Port Name List attribute shall be a zero length Attached Port Name List.

6.8.3.6.2.6 Logical Port Object Port State attribute

The values of the Logical Port Object Port State attribute shall be as shown in table 483.

Table 483 – Port State encoding

Encoded value	Description
00h	Unknown
01h	Online - a frame may be passed through the FC_Port
02h	Offline - a frame is not able to be passed through the FC_Port
E0h to FFh	Vendor specific
all others	Reserved

This standard does not define how this attribute is registered with the Enhanced Fabric Configuration Server. The null Port State attribute value is set to ‘Unknown’.

6.8.3.6.2.7 Logical Port Object Zoning Enforcement Status attribute

The format of the Zoning Enforcement Status attribute is depicted in table 484.

Table 484 – Zoning Enforcement Status

Item	Size (Bytes)
F_Port_Name	8
Port enforcement status	4

F_Port_Name: This field contains the F_Port_Name of the Fx_Port that the enforcement status object is referencing.

Port Enforcement Status: This is a 32 bit wide bit field that reports the actual enforcement status of the named Fx_Port. The defined bits are depicted in table 485.

Table 485 – Port Enforcement Status Bit Definitions

Bit	Interpretation
0	1 = Soft Zoning enforcement on 0 = Soft Zoning enforcement off
1	1 = Hard Zoning enforcement on 0 = Hard Zoning enforcement off
2	1 = Broadcast Zoning Enforcement on 0 = Broadcast Zoning Enforcement off
all others	Reserved

6.8.3.6.2.8 Logical Port Object Physical Port Correlatable Identifier attribute

The Physical Port Object Correlatable Identifier (see 6.8.3.7.2.2) that the Logical Port Object is associated with.

6.8.3.6.2.9 Logical Port Object Discriminator attribute

The values of the Logical Port Object Discriminator attribute shall be as shown in table 486.

Table 486 – Discriminator encoding

Encoded value	Description
00h	Unknown
01h	Native FC
02h	FCoE
03h	VFT
04h	FCIP
all others	Reserved

This standard does not define how this attribute is registered with the Enhanced Fabric Configuration Server. The null Discriminator attribute value is set to ‘Unknown’.

6.8.3.7 Physical Port Object

6.8.3.7.1 Physical Port Object Attribute Block

6.8.3.7.1.1 Overview

The Physical Port Object Attribute Block is a variable length structure that contains attributes registered for the specified Physical Port Object. The format of the Physical Attribute Block is depicted in table 487.

Table 487 – Physical Port Object Attribute Block

Item	Size (Bytes)
Number of Physical Port Object Attribute Entries (n)	4
Enhanced Fabric Configuration Server Attribute Entry 1	see table 488
Enhanced Fabric Configuration Server Attribute Entry 2	see table 488
...	
Enhanced Fabric Configuration Server Attribute Entry n-1	see table 488
Enhanced Fabric Configuration Server Attribute Entry n	see table 488

6.8.3.7.1.2 Number of Physical Port Object Attribute Entries

This field specifies the number of Physical Port Object Attribute Entries contained in the Physical Port Object Attribute Block. This value shall be greater than or equal to one.

6.8.3.7.1.3 Enhanced Fabric Configuration Server Attribute Entry

An Enhanced Fabric Configuration Server Attribute Entry specifies a particular attribute registered with a Physical Port Object.

6.8.3.7.2 Physical Port Object Attributes

6.8.3.7.2.1 Overview

The Physical Port Object Attribute Entry Types and their associated values are depicted in table 488.

Table 488 – Physical Port Object Attribute Entry Types and their associated Values

Tag	Value				
	Description	Length (Bytes)	Data Type	Required	Multiples allowed[a]
0001h	Correlatable Identifier	1 to 255	ASCII	Yes	No
0002h	Transport Type	1	Binary	Yes	No
0003h	Name	8	Binary	No	Yes
0004h	TX Type	1	Binary	Yes	No
0005h	Module Type	1	Binary	No	No
0006h	Physical Port Number	4	Binary	No	Yes
0007h	Port State	2	Binary	Yes	No
0008h	Port Speed Capabilities	4	Binary	Yes	No
0009h	Port Operating Speed	4	Binary	Yes	No
0010h	Physical Location	1 to 255	ASCII	No	No
0011h	Parallel Lane Supported	1	Binary	Yes	No
0012h	Parallel Lane Operational	1	Binary	Yes	No
0013h	Parallel Lane Only	1	Binary	Yes	No
other values	Reserved				

[a] If a Physical Port Attribute Block contains multiple types for a type that does not allow multiples the command shall be rejected with a reason code of ‘Unable to perform command request’ and a reason code explanation of “Physical Port Object Attribute Block Contains Multiple Attributes of the Same Type”.

6.8.3.7.2.2 Physical Port Object Correlatable Identifier attribute

A printable ASCII string, terminated with a null (00h), that uniquely identifies the Physical Port Object.

This standard does not define how this attribute is registered with the Enhanced Fabric Configuration Server. The contents of the Physical Port Object Correlatable Identifier shall not be restricted by the Enhanced Fabric Configuration Server.

6.8.3.7.2.3 Physical Port Object Transport Type attribute

The values of the Physical Port Object Transport Type attribute shall be as shown in table 489.

Table 489 – Physical Port Object Transport Types

Encoded value	Description
00h	Unknown
01h	FC
02h	Ethernet
03h	IP
E0h to FFh	Vendor specific
all others	Reserved

6.8.3.7.2.4 Physical Port Object Name attribute

The format of the Name attribute shall be identical to the Name_Identifier format. The value of the Name attribute shall be the same as the value Port_Name in the Fabric Login ELS Accept payload.

This standard does not define how this attribute is registered with the Enhanced Fabric Configuration Server. The null value for the Port Name attribute is 00 00 00 00 00 00 00 00h.

6.8.3.7.2.5 Physical Port Object TX Type attribute

This attribute describes the technology that is incorporated in the transmitter of the module. The values of the TX Type attribute shall be as shown in table 490.

Table 490 – TX Type encoding

Encoded value	Description
01h	Unknown
02h	Long wave laser - LL (1 550 nm)
03h	Short wave laser - SN (850 nm)
04h	Long wave laser cost reduced - LC (1 310 nm)
05h	Electrical - EL
06h	10GBASE-SR 850nm laser[a]
07h	10GBASE-LR 1310nm laser[a]
08h	10GBASE-ER 1550nm laser[a]
09h	10GBASE-LX4 WWDM 1300nm laser[a]
0Ah	10GBASE-SW 850nm laser[a]
0Bh	10GBASE-LW 1310nm laser[a]
0Ch	10GBASE-EW 1550nm laser[a]
0Dh	10GBASE-CX4[a]
0Fh	Long wave laser - LZ (1 490 nm)
all others	Reserved

[a] See IEEE 802.3-2005.

This standard does not define how this attribute is registered with the Enhanced Fabric Configuration Server. The null TX Type attribute value is set to ‘Unknown’.

6.8.3.7.2.6 Physical Port Object Module Type attribute

This attribute describes the form factor of the module. The values of the Physical Port Module Type attribute shall be as shown in table 491.

Table 491 – Module Type encoding

Encoded value	Description
01h	Unknown
02h	Other
03h	Obsolete
04h	Embedded
05h	GLM
06h	GBIC with serial ID
07h	GBIC without serial ID
08h	SFP with Serial ID
09h	SFP without Serial ID
0Ah	XFP
0Bh	X2 Short
0Ch	X2 Medium
0Dh	X2 Tall
0Eh	XPAK Short

Table 491 – Module Type encoding(Continued)

Encoded value	Description
0Fh	XPAK Medium
10h	XPAK Tall
11h	XENPAK
12h	SFP-DWDM
13h	QSFP
14h	X2-DWDM
15h	CFP
16h	CFP2
17h	CFP4
18h	QSFP+
19h	QSFP28
all others	Reserved

This standard does not define how this attribute is registered with the Enhanced Fabric Configuration Server. The null Port Module Type attribute value is set to ‘Unknown’.

6.8.3.7.2.7 Physical Port Object Physical Port Number

The Physical Port Number is the value used with the Domain_ID that identifies a Zone Member Identifier Type 2 (i.e. Domain_ID and physical port).

The format of the Physical Port Object Physical Port Number attribute shall be as shown in table 492. The contents of this field are not defined and shall not be restricted by the Enhanced Fabric Configuration Server, due to vendor specific methods for numbering physical ports.

Table 492 – Physical Port Number Format

Item	Size (Bytes)
Physical Port Number	4

This standard does not define how this attribute is registered with the Enhanced Fabric Configuration Server. The null value for the Physical Port Number attribute is 00 00 00 00h.

6.8.3.7.2.8 Physical Port Object Port State attribute

The values of the Physical Port Object Port State attribute shall be as shown in table 493.

Table 493 – Physical Port Object Port State attribute

Encoded value	Description
00h	Unknown
01h	Online - a frame may be passed through the FC_Port
02h	Offline - a frame is not able to be passed through the FC_Port
03h	Testing - FC_Port is in a test state
04h	Fault - FC_Port is not operational
E0h to FFh	Vendor specific
all others	Reserved

This standard does not define how this attribute is registered with the Enhanced Fabric Configuration Server. The null Port State attribute value is set to ‘Unknown’.

6.8.3.7.2.9 Physical Port Object Port Speed Capabilities attribute

The Port Speed Capabilities attribute identifies the data transfer rate capabilities of the LCF within the FC_Port. The format of the Port Speed Capabilities attribute shall be as shown in table 494.

Table 494 – Port Speed Capabilities attribute format

Item	Size (Bytes)
Port Speed Capabilities	4

All the LCF's potential supported data transfer speeds are indicated by setting the appropriate bit in the Speed Capabilities field to one. More than one bit may be set at a time. Valid bits are as shown in table 495.

Table 495 – Port Speed Capabilities field bits

Bit Position	Description	Bit Position	Description
31	1 GFC	15	50 GE
30	2 GFC	14	200 GE
29	4 GFC	13	400 GE
28	10 GFC[a]	12 to 1	Reserved
27	8 GFC		
26	16 GFC		
25	20 GFC		
24	32 GFC		
23	40 GFC[a]		
22	10 GE		
21	40 GE		
20	100 GE		
19	128 GFC		
18	25 GE		
17	64 GFC		
16	256 GFC	0	Unknown

[a] Legacy implementations may have used this bit for Ethernet.

6.8.3.7.2.10 Physical Port Object Port Operating Speed attribute

The Port Operating Speed attribute identifies the current operating data transfer rate of the LCF within an FC_Port. The format of the Port Operating Speed attribute shall be as in table 496.

Table 496 – Port Operating Speed attribute format

Item	Size (Bytes)
Port Operating Speed	4

When a bit is set to one, it indicates the LCF is operating at the designated speed. Only one bit shall be set at a time. If the operating speed has not been established, then the “Speed not established” bit is set to one. If the LCF’s operating speed isn’t identifiable, then the “Unknown” bit is set to one. Valid bits are as shown in table 497.

Table 497 – Port Operating Speed field bits

Bit Position	Description	Bit Position	Description
31	1 GFC	15	50 GE
30	2 GFC	14	200 GE
29	4 GFC	13	400 GE
28	10 GFC[a]	12 to 2	Reserved
27	8 GFC		
26	16 GFC		
25	20 GFC		
24	32 GFC		
23	40 GFC[a]		
22	10 GE		
21	40 GE		
20	100 GE		
19	128 GFC		
18	25 GE		
17	64 GFC	1	Unknown
16	256 GFC	0	Speed not established

[a] Legacy implementations may have used this bit for Ethernet.

6.8.3.7.2.11 Physical Port Object Physical Location attribute

The Physical Location is a printable ASCII string, terminated with a null (00h), that is a label for a port supporting administrative identification.

This standard does not define how the attribute is registered with the Enhanced Fabric Configuration Server.

6.8.3.7.2.12 Physical Port Object Parallel Lane Supported attribute

If the Physical Port Object Parallel Lane Supported attribute is set to one, then:

a) this physical port supports parallel lanes (see FC-FS-4); and

b) the Physical Port Object Parallel Lane Operational attribute and the Physical Port Object Parallel Lane Only attribute are used to indicate the parallel lane operational mode of this physical port.

If the Physical Port Object Parallel Lane Supported attribute is set to zero, then:

a) this physical port does not support parallel lanes; and

b) the Physical Port Object Parallel Lane Operational attribute and the Physical Port Object Parallel Lane Only attribute are not meaningful.

The format of the Physical Port Object Parallel Lane Supported attribute is specified in table 498.

Table 498 – Physical Port Object Parallel Lane Supported attribute

Encoded value	Description
0h	Unsupported
1h	Supported
all others	Reserved

This standard does not define how this attribute is registered with the Enhanced Fabric Configuration Server.

6.8.3.7.2.13 Physical Port Object Parallel Lane Operational attribute

If the Physical Port Object Parallel Lane Operational attribute is set to one, then this physical port is operating in parallel lane mode (i.e., operating with one or more other lanes as a link, see FC-FS-4).

If the Physical Port Object Parallel Lane Operational attribute is set to zero, then this physical port is operating in individual lane mode (i.e., operating with no other lanes as a link).

The format of the Physical Port Object Parallel Lane Operational attribute is specified in table 499.

Table 499 – Physical Port Object Parallel Lane Operational attribute

Encoded value	Description
0h	Not operational
1h	Operational
all others	Reserved

This standard does not define how this attribute is registered with the Enhanced Fabric Configuration Server.

6.8.3.7.2.14 Physical Port Object Parallel Lane Only attribute

If the Physical Port Object Parallel Lane Only attribute is set to one, then this physical port is only able to operate in parallel lane mode.

If the Physical Port Object Parallel Lane Only attribute is set to zero, then this physical port may operate in either individual lane mode or parallel lane mode.

The format of the Physical Port Object Parallel Lane Only attribute is specified in table 500.

Table 500 – Physical Port Object Parallel Lane Only attribute

Encoded value	Description
0h	Individual lane mode or parallel lane mode
1h	Parallel lane only
all others	Reserved

This standard does not define how this attribute is registered with the Enhanced Fabric Configuration Server.

6.8.4 Reason code explanations

A Reject CT_IU (see 4.4.4) shall notify the requestor that the request has been unsuccessfully completed. The first error condition encountered shall be the error reported by the Reject CT_IU.

If a valid Enhanced Fabric Configuration Server request is not received, the request is rejected with a reason code of “Invalid Command code” and a reason code explanation of “No additional explanation”.

If an Enhanced Fabric Configuration Server request is rejected with a reason code of ‘Unable to perform command request’, then one of the reason code explanations, shown in table 501, is returned.

Table 501 – Reject CT_IU Reason Code Explanations

Encoded value	Description
00h	No additional explanation
01h	Invalid Name_Identifier for Interconnect Element or Port
02h	Invalid Physical Object Correlatable Identifier
10h	Interconnect Element Object List not available
11h	Interconnect Element Object Port List not available
12h	Physical Object Port List not available

Table 501 – Reject CT_IU Reason Code Explanations(Continued)

Encoded value	Description
20h	Fabric Object not available
21h	Interconnect Element Object not available
22h	Physical Object not available
23h	Logical Port Object not available
24h	Physical Port Object not available
30h	Attached Port Name List not available
40h	Unable to register Interconnect Element Logical Name
41h to 6Fh	Vendor Specific
Others	Reserved

If an Enhanced Fabric Configuration Server Query request other than eGIEL and GPL is rejected by the Enhanced Fabric Configuration Server because the attribute specified in the request is not found in the Enhanced Fabric Configuration Server data base, then the Reject CT_IU reason code shall be ‘Unable to perform command request’, with a reason code explanation that indicates the specified attribute is not available.

6.8.5 Commands

6.8.5.1 Overview

The commands defined for the Enhanced Fabric Configuration Server are summarized in table 465.

6.8.5.2 Query - Get Interconnect Element List (eGIEL)

The Enhanced Fabric Configuration Server shall, if it receives a eGIEL request, return all Interconnect Element Object Names in the Fabric. The format of the eGIEL Request CT_IU is shown in table 502.

Table 502 – eGIEL Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

If the eGIEL request is not supported a Reject CT_IU should be returned and the reason code should be ‘Command Not Supported’, no further explanation.

The format of the Accept CT_IU to a eGIEL request is shown in table 503.

Table 503 – Accept CT_IU to eGIEL Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Number of Interconnect Element entries (n)	4
Interconnect Element Name #1	8
Interconnect Element Name #2	8
...	
Interconnect Element Name #n	8

One or more Interconnect Element Name entries are returned, and the Interconnect Element Name entries may be returned in any order, and the order may be different for every request even if the same Interconnect Element Name entries are returned and the requestor is the same.

6.8.5.3 Query - Get Interconnect Element Attribute Block (eGIEAB)

The Enhanced Fabric Configuration Server shall, if it receives a eGIEAB request, return the Interconnect Element Object Attribute Block for the specified Interconnect Element Name. The format of the eGIEAB Request CT_IU is shown in table 504.

Table 504 – eGIEAB Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Interconnect Element Name	8

If the eGIEAB request is not supported a Reject CT_IU should be returned and the reason code should be ‘Command Not Supported’, no further explanation.

The format of the Accept CT_IU to a eGIEAB request is shown in table 505.

Table 505 – Accept CT_IU to eGIEAB Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Interconnect Element Object Attribute Block	see 6.8.3.4.1

6.8.5.4 Query - Get Interconnect Element Port List (eGIEPL)

The Enhanced Fabric Configuration Server shall, if it receives a eGIEPL request, return all Logical Port Names for the specified Interconnect Element Name. The format of the eGIEPL Request CT_IU is shown in table 506.

Table 506 – eGIEPL Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Interconnect Element Name	8

If the eGIEPL request is not supported a Reject CT_IU should be returned and the reason code should be ‘Command Not Supported’, no further explanation.

The format of the Accept CT_IU to a eGIEPL request is shown in table 507.

Table 507 – Accept CT_IU to eGIEPL Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Number of Logical Port Name entries (n)	4
Logical Port Name #1	8
Logical Port Name #2	8
...	
Logical Port Name #n	8

One or more Logical Port Names are returned, and the Logical Port Names may be returned in any order, and the order may be different for every request even if the same Logical Port Names are returned and the requestor is the same.

6.8.5.5 Query - Get Fabric Object (eGFO)

The Enhanced Fabric Configuration Server shall, if it receives a eGFO request, return the Fabric Object. The format of the eGFO Request CT_IU is shown in table 508.

Table 508 – eGFO Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

If the eGFO request is not supported a Reject CT_IU should be returned and the reason code should be ‘Command Not Supported’, no further explanation.

The format of the Accept CT_IU to a eGFO request is shown in table 509.

Table 509 – Accept CT_IU to eGFO Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Fabric Object	see 6.8.3.3

6.8.5.6 Query - Get Physical Object Attribute Block (eGPOAB)

The Enhanced Fabric Configuration Server shall, if it receives a eGPOAB request, return the Physical Object Attribute Block for the specified Physical Object Correlatable Identifier. The format of the eGPOAB Request CT_IU is shown in table 510.

Table 510 – eGPOAB Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Physical Object Correlatable Identifier	256

If the eGPOAB request is not supported a Reject CT_IU should be returned and the reason code should be ‘Command Not Supported’, no further explanation.

The format of the Accept CT_IU to a eGPOAB request is shown in table 511.

Table 511 – Accept CT_IU to eGPOAB Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Physical Object Attribute Block	see 6.8.3.5.1

6.8.5.7 Query - Get Physical Object Port List (eGPOPL)

The Enhanced Fabric Configuration Server shall, if it receives a eGPOPL request, return all Physical Port Object Correlatable Identifiers for the specified Physical Object Correlatable Identifier. The format of the eGPOPL Request CT_IU is shown in table 512.

Table 512 – eGPOPL Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Physical Object Correlatable Identifier	256

If the eGPOPL request is not supported a Reject CT_IU should be returned and the reason code should be ‘Command Not Supported’, no further explanation.

The format of the Accept CT_IU to a eGPOPL request is shown in table 513.

Table 513 – Accept CT_IU to eGPOPL Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Number of Physical Port Object Correlatable Identifier entries (n)	4
Physical Port Object Correlatable Identifier #1	see 6.8.3.7.2.2
Physical Port Object Correlatable Identifier #2	see 6.8.3.7.2.2
...	
Physical Port Object Correlatable Identifier #n	see 6.8.3.7.2.2

One or more Physical Port Object Correlatable Identifier entries are returned, and the Physical Port Object Correlatable Identifier entries may be returned in any order, and the order may be different for every request even if the same Physical Port Object Correlatable Identifier entries are returned and the requestor is the same.

6.8.5.8 Query - Get Logical Port Attribute Block (eGLPAB)

The Enhanced Fabric Configuration Server shall, if it receives a eGLPAB request, return the Logical Port Object Attribute Block for the specified Logical Port Name. The format of the eGLPAB Request CT_IU is shown in table 514.

Table 514 – eGLPAB Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Logical Port Name	8

If the eGLPAB request is not supported a Reject CT_IU should be returned and the reason code should be ‘Command Not Supported’, no further explanation.

The format of the Accept CT_IU to a eGLPAB request is shown in table 515.

Table 515 – Accept CT_IU to eGLPAB Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Logical Port Object Attribute Block	see 6.8.3.6.1

6.8.5.9 Query - Get Attached Port List (eGAPL)

The Enhanced Fabric Configuration Server shall, if it receives a eGAPL request, return all Logical Port Object Names for the specified Logical Port Name. The format of the eGAPL Request CT_IU is shown in table 516.

Table 516 – eGAPL Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Logical Port Name	8

If the eGAPL request is not supported a Reject CT_IU should be returned and the reason code should be ‘Command Not Supported’, no further explanation.

The format of the Accept CT_IU to a eGAPL request is shown in table 517.

Table 517 – Accept CT_IU to eGAPL Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Number of Logical Port Name entries (n)	4
Logical Port Name #1	8
Logical Port Name #2	8
...	
Logical Port Name #n	8

One or more Logical Port Name entries are returned, and the Logical Port Name entries may be returned in any order, and the order may be different for every request even if the same Logical Port Name entries are returned and the requestor is the same.

6.8.5.10 Query - Get Physical Port Object Attribute Block (eGPPAB)

The Enhanced Fabric Configuration Server shall, if it receives a eGPPAB request, return the Physical Port Object Attribute Block for the specified Physical Port Object Correlatable Identifier. The format of the eGPPAB Request CT_IU is shown in table 518.

Table 518 – eGPPAB Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Physical Port Object Correlatable Identifier	see 6.8.3.7.2.2

If the eGPPAB request is not supported a Reject CT_IU should be returned and the reason code should be ‘Command Not Supported’, no further explanation.

The format of the Accept CT_IU to a eGPPAB request is shown in table 519.

Table 519 – Accept CT_IU to eGPPAB Request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Physical Port Object Attribute Block	see 6.8.3.7.1

6.8.5.11 Register Interconnect Element Logical Name (eRIELN)

The eRIELN Enhanced Fabric Configuration Server request shall be used to associate a Logical Name with a given Interconnect Element Object.

The Enhanced Fabric Configuration Server shall not attempt validation of the Logical Name attribute. This means that any Logical Name value shall be accepted.

Deregistration may be accomplished by registering a null Logical Name 6.8.3.4.2.6.

If the eRIELN request is not supported a Reject CT_IU should be returned and the reason code should be ‘Command Not Supported’, no further explanation.

The format of the eRIELN Request CT_IU is shown in table 520.

Table 520 – eRIELN Request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Interconnect Element Name	8
Interconnect Element Logical Name	256

The format of the eRIELN Accept CT_IU is shown in table 521.

Table 521 – eRIELN Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

6.9 Application Server

6.9.1 Overview

The Application Server provides a way to manage application specific services (e.g., application identifiers).

Requests for the Application Server are carried over the Common Transport (see clause 4).

The Application Server is intended to be distributed among Switches, making the Application Server immediately available to an Nx_Port once it has successfully completed Fabric Login. However, the Application Server is not restricted or required to be part of a Fabric, and may be located in any Nx_Port.

6.9.2 Protocol

6.9.2.1 Overview

Application Server registration, deregistration, and queries are managed through protocols containing a set of request CT_IUs and response CT_IUs supported by the Application Server.

For an Application Server request, the payload shall be transported from the requestor to the Application Server using a request CT_IU. The corresponding Application Server response is transported from the Application Server to the requestor, in the Exchange established by the requestor, using a response CT_IU.

The action of the Application Server is unaffected by Server Sessions.

6.9.2.2 CT_IU preamble values

The following values shall be set in the CT_IU preamble for Application Server request and their responses, and fields not specified here shall be set as specified in 4.3.2:

a) GS_Subtype: as indicated in table 125; and

b) Command Code: see table 522 for request command codes.

Table 522 – Application Server - request command codes

Code	Mnemonic	Description	Reference subclause
0100h	GAPPIA_ENT	Get Application Identifier Allocations - Entity Identifier	6.9.5.2
0101h	GALLAPPIA	Get All Application Identifier Allocations	6.9.5.3
0102h	GALLAPPIA_ID	Get All Application Identifier Allocations - N_Port_ID	6.9.5.4
0103h	GAPPIA_IDAPP	Get Application Identifier Allocations - N_Port_ID and Application Identifier	6.9.5.5
0104h	GENTATTR_IDAPP	Get Entity Attributes - N_Port_ID and Application Identifier	6.9.5.6
0105h	GALLENTATTR_ID	Get Entity Attributes - N_Port_ID	6.9.5.7
0106h	GENTATTR_ENT	Get Entity Attributes - Entity Identifier	6.9.5.8
0107h	GALLENTATTR	Get Entity Attributes - All	6.9.5.9
0200h	RAPP_IDENT	Register Application Identifier - N_Port_ID and Entity Identifier	6.9.5.10
0201h	RENTATTR_APPIA	Register Entity Attributes - Application Identifier Allocation	6.9.5.11
0300h	DAPP_IDENT	Deregister Application Identifier - N_Port_ID and Entity Identifier	6.9.5.12
0301h	DALLAPP_ID	Deregister All Application Identifiers - N_Port_ID	6.9.5.13
other	reserved		

6.9.2.3 Registration

The registration requests defined for the Application Server are summarized in table 522. This standard does not define the registration of attributes which do not have a corresponding registration request.

The Application Server may reject registrations due to Application Server resource limitations. However, the Application Server shall support registration of all attributes, once registration of a single attribute has been accepted.

The Application Server may reject any registration requests for reasons not specified in this document.

If overlapping registrations for the same attribute are performed, then the Application Server shall, when all registrations have been completed, leave the attribute as one of the registered attribute values. However, it is indeterminate which of the overlapping registration requests take precedence.

6.9.2.4 Queries

The Application Server may reject any query request for reasons not specified in this standard. The queries defined for the Application Server are summarized in table 522.

6.9.3 Application Server objects and attributes

6.9.3.1 Application Identifier Allocation object

The Application Identifier Allocation object is a structure that contains application identifier to entity identifier allocations. The format of the Application Identifier Allocation object is specified in table 523.

Table 523 – Application Identifier Allocation object

Item	Size (Bytes)
Reserved	1
N_Port_ID	3
Application Identifier	4
Entity Identifier object	see 6.9.3.2

N_Port_ID: contains an N_Port_ID associated with the Application Identifier Allocation object.

Application Identifier: contains a value allocated to the specified entity identifier. Valid application identifier values are specified in table 524.

Table 524 – Application identifier values

Encoded value	Description
0000 0000h	Unknown application identifier
0000 0001h	Base application identifier - the identifier for the base entity (e.g., a hypervisor).
0000 0002h to 0000 000Fh	Reserved
0000 0010h to FFFF FFFFh	Application identifiers

6.9.3.2 Entity Identifier object

The Entity Identifier object is a structure that contains an entity identifier.The format of the Entity Identifier object is specified in table 525.

Table 525 – Entity Identifier object

Item	Size (Bytes)
Entity Identifier Length	1
Entity Identifier	n
Reserved	255-n

Entity Identifier Length: contains the length of the Entity Identifier field.

Entity Identifier: contains an identifier for the entity (e.g., a UUID value (see RFC 4122)).

6.9.3.3 Entity Attribute Allocation object

The Entity Attribute Allocation object is a structure that contains application identifier allocation object to entity attribute allocations. The format of the Entity Attribute Allocation object is specified in table 526.

Table 526 – Entity Attribute Allocation object

Item	Size (Bytes)
Application Identifier Allocation object	see 6.9.3.1
Reserved	3
Number of Entity Attribute Entries	1
Entity Attribute Entry #1	see 6.9.3.4
Entity Attribute Entry #n	see 6.9.3.4

Application Identifier Allocation object: contains an object specifying the Entity identifier associated with the list of Entity attributes.

Number of Entity Attribute Entries: contains a count of the entity attributes associated with the Entity identified by the Application Identifier.

Entity Attribute Entry: contains an attribute for the entity (e.g., an Entity Name).

6.9.3.4 Entity Attribute format

The Application Server defines a general format to be used for attributes associated with Entity Identifier objects. The general format of the Entity Attribute is depicted in

Table 527 – Entity Attribute Entry

Item	Size (Bytes)
Entity Attribute Type	2
Entity Attribute Length	2
Entity Attribute Value	(see table 528)

Entity Attribute Type: This field indicates the Entity Attribute Type. Valid Attribute types are specific to the object to which they are associated. The Type codes are defined in

Entity Attribute Length: This field indicates the total length of the Entity Attribute. The total length in bytes shall be a multiple of four and includes the Entity Attribute Type, Entity Attribute Length, and Entity Attribute Value fields.

Entity Attribute Value: This field specifies the Entity Attribute Value. Entity Attribute Values shall be at least four bytes in length and the length shall be a multiple of four. For variable length Attribute Value fields, fill bytes are added as necessary to the end of the actual value in order to ensure that the length of the value field is a multiple of four. Fill bytes shall be 00h. Entity Attribute types are defined in table 528.

Table 528 – Entity Attribute Types and their associated Values

Tag	Value				
	Description	Length (Bytes)	Type	Required	Multiples allowed[a]
0001h	Entity Name	1 to 63	ASCII	No	No
0002h	Host Identifier	1 to 63	ASCII	No	No
0003h	Symbolic Data	1 to 127	ASCII	No	No
other values	Reserved				

[a] If an Entity Attribute registration contains multiple types for a type that does not allow multiples, the command shall be rejected with a reason code of ‘Unable to perform command request’ and a reason code explanation of “Entity Attribute Registration Contains Multiple Attributes of the Same Type”.

Entity Name attribute: The Entity Name Attribute contains a null (i.e. 00h) terminated printable ASCII string that specifies an alias name for the Entity. This attribute is registered with the Application Server as an Entity Identifier attribute. The registration of this attribute is optional.

Host Identifier attribute: The Host Identifier Attribute contains a null (i.e. 00h) terminated printable ASCII string that specifies the common host name associated with the Entity. This attribute is registered with the Application Server as an Entity Identifier attribute. The registration of this attribute is optional.

Symbolic Data attribute: The Symbolic Data Attribute contains a null (i.e. 00h) terminated printable ASCII string that specifies the common host name associated with the Entity. This attribute is registered with the Application Server as an Entity Identifier attribute. The registration of this attribute is optional.

6.9.4 Reason code explanations

A reject CT_IU (see 4.4.4) shall notify the requestor that the request has been unsuccessfully completed. The first error condition encountered shall be the error reported by the reject CT_IU.

If a valid Application Server request is not received, the request is rejected with a reason code of “Invalid Command code” and a reason code explanation of “No additional explanation”.

If an Application Server request is rejected with a reason code of ‘Unable to perform command request’ or ‘Protocol error’, then one of the reason code explanations, specified in table 529, is returned.

Table 529 – Reject CT_IU reason code explanations

Encoded value	Description
00h	No additional explanation
10h	Access denied
11h	Entity attribute registration contains multiple attributes of the same type
29h	Insufficient resources available
40h	Application Identifier Allocation object(s) not available
Others	Reserved

If an Application Server Query request is rejected by the Application Server because the attribute(s) specified in the request is/are not found in the Application Server data base, then the reject CT_IU reason code shall be ‘Unable to perform command request’, with a reason code explanation that indicates the specified attribute is not available.

6.9.5 Commands

6.9.5.1 Overview

The commands defined for the Application Server are summarized in table 522.

6.9.5.2 Query - Get Application Identifier Allocation (GAPPIA_ENT)

The Application Server shall, if it receives a GAPPIA_ENT request, return all requested Application Identifier Allocation objects in the Fabric for the specified entity identifiers. The format of the GAPPIA_ENT request CT_IU is specified in table 530.

Table 530 – GAPPIA_ENT request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Control (0 r r r r r r r)	1
Reserved	3
Entity Identifier object #1	see 6.9.3.2
...	
Control (1 r r r r r r r)	1
Reserved	3
Entity Identifier object #n	see 6.9.3.2

One or more Entity Identifier objects are specified. Each specified Entity Identifier object is preceded by an 8 bit Control field. The format of the Control field is:

a) Bit 7 is set to zero if the Entity Identifier object following the Control field is not the last Entity Identifier object to be specified by the request CT_IU. Bit 7 is set to one if the Entity Identifier object following the Control field is the last Entity Identifier object specified by the accept CT_IU; and

b) Bits 6 to 0 are reserved.

If the GAPPIA_ENT request is not supported a reject CT_IU shall be returned with reason code ‘Command not supported’ and reason code explanation ‘No further explanation’.

The format of the accept CT_IU to a GAPPIA_ENT request is specified in table 531.

Table 531 – Accept CT_IU to GAPPIA_ENT request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Control (0 r r r r r r r)	1
Reserved	3
Application Identifier Allocation object #1	see 6.9.3.1
...	
Control (1 r r r r r r r)	1
Reserved	3
Application Identifier Allocation object #n	see 6.9.3.1

One or more Application Identifier Allocation objects are returned, and the number of Application Identifier Allocation objects returned may be greater than the number of Entity Identifiers specified in the request. Each returned Application Identifier Allocation object is preceded by an 8 bit Control field. The format of the Control field is:

a) Bit 7 is set to zero if the Application Identifier Allocation object following the Control field is not the last Application Identifier Allocation object to be returned by the accept CT_IU. Bit 7 is set to one if the Application Identifier Allocation object following the Control field is the last Application Identifier Allocation object returned by the accept CT_IU; and

b) Bits 6 to 0 are reserved.

If the entity identifier specified in the Application Identifier Allocation object does not have an application identifier assignment, then the N_Port_ID field is set to all zeroes and the Application Identifier field value is set to ‘Unknown application identifier’ (see table 524) in the Application Identifier Allocation object.

6.9.5.3 Query - Get All Application Identifier Allocations (GALLAPPIA)

The Application Server shall, if it receives a GALLAPPIA request, return all Application Identifier Allocation objects. The format of the GALLAPPIA request CT_IU is specified in table 532.

Table 532 – GALLAPPIA request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

If the GALLAPPIA request is not supported a reject CT_IU shall be returned with reason code ‘Command not supported’ and reason code explanation ‘No further explanation’.

The format of the accept CT_IU to a GALLAPPIA request is specified in table 533.

Table 533 – Accept CT_IU to GALLAPPIA request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Control (0 r r r r r r r)	1
Reserved	3
Application Identifier Allocation object #1	see 6.9.3.1
...	
Control (1 r r r r r r r)	1
Reserved	3
Application Identifier Allocation object #n	see 6.9.3.1

One or more Application Identifier Allocation objects are returned. Each returned Application Identifier Allocation object is preceded by an 8 bit Control field. The format of the Control field is:

a) Bit 7 is set to zero if the Application Identifier Allocation object following the Control field is not the last Application Identifier Allocation object to be returned by the accept CT_IU. Bit 7 is set to one if the Application Identifier Allocation object following the Control field is the last Application Identifier Allocation object returned by the accept CT_IU; and

b) Bits 6 to 0 are reserved.

6.9.5.4 Query - Get All Application Identifier Allocations (GALLAPPIA_ID)

The Application Server shall, if it receives a GALLAPPIA_ID request, return all Application Identifier Allocation objects associated with the N_Port_ID of the requestor. The format of the GALLAPPIA_ID request CT_IU is specified in table 534.

Table 534 – GALLAPPIA_ID request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Reserved	1
N_Port_ID	3

If the GALLAPPIA_ID request is not supported a reject CT_IU shall be returned with reason code ‘Command not supported’ and reason code explanation ‘No further explanation’.

The format of the accept CT_IU to a GALLAPPIA_ID request is specified in table 535.

Table 535 – Accept CT_IU to GALLAPPIA_ID request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Control (0 r r r r r r r)	1
Reserved	3
Application Identifier Allocation object #1	see 6.9.3.1
...	
Control (1 r r r r r r r)	1
Reserved	3
Application Identifier Allocation object #n	see 6.9.3.1

One or more Application Identifier Allocation objects are returned. Each returned Application Identifier Allocation object is preceded by an 8 bit Control field. The format of the Control field is:

a) Bit 7 is set to zero if the Application Identifier Allocation object following the Control field is not the last Application Identifier Allocation object to be returned by the accept CT_IU. Bit 7 is set to one if the Application Identifier Allocation object following the Control field is the last Application Identifier Allocation object returned by the accept CT_IU; and

b) Bits 6 to 0 are reserved.

6.9.5.5 Query - Get Application Identifier Allocation (GAPPIA_IDAPP)

The Application Server shall, if it receives a GAPPIA_IDAPP request, return Application Identifier Allocation objects for the specified N_Port_ID and application identifier pairs. The format of the GAPPIA_IDAPP request CT_IU is specified in table 536.

Table 536 – GAPPIA_IDAPP request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Control (0 r r r r r r r)	1
N_Port_ID #1	3
Application Identifier #1	4
...	
Control (1 r r r r r r r)	1
N_Port_ID #n	3
Application Identifier #n	4

One or more N_Port_ID field and Application Identifier field pairs are specified. Each specified N_Port_ID field and Application Identifier field pair is preceded by an 8 bit Control field. The format of the Control field is:

a) Bit 7 is set to zero if the N_Port_ID field and Application Identifier field pair following the Control field is not the last N_Port_ID field and Application Identifier field pair to be specified by the request CT_IU. Bit 7 is set to one if the N_Port_ID field and Application Identifier field pair following the Control field is the last N_Port_ID field and Application Identifier field pair specified by the accept CT_IU; and

b) Bits 6 to 0 are reserved.

If the GAPPIA_IDAPP request is not supported a reject CT_IU shall be returned with reason code ‘Command not supported’ and reason code explanation ‘No further explanation’.

The format of the accept CT_IU to a GAPPIA_IDAPP request is specified in table 537.

Table 537 – Accept CT_IU to GAPPIA_IDAPP request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Control (0 r r r r r r r)	1
Reserved	3
Application Identifier Allocation object #1	see 6.9.3.1
...	
Control (1 r r r r r r r)	1
Reserved	3
Application Identifier Allocation object #n	see 6.9.3.1

One or more Application Identifier Allocation objects are returned. Each returned Application Identifier Allocation object is preceded by an 8 bit Control field. The format of the Control field is:

a) Bit 7 is set to zero if the Application Identifier Allocation object following the Control field is not the last Application Identifier Allocation object to be returned by the accept CT_IU; the bit is set to one if the Application Identifier Allocation object following the Control field is the last Application Identifier Allocation object returned by the accept CT_IU; and

b) Bits 6 to 0 are reserved.

If an N_Port_ID and application identifier pair specified in the request does not have an application identifier allocation, then the Entity Identifier field value in the Application Identifier Allocation object is set to all zeros.

6.9.5.6 Query - Get Entity Attributes (GENTATTR_IDAPP)

The Application Server shall, if it receives a GENTATTR_IDAPP request, return all the Entity Attribute Allocation objects for the specified N_Port_ID and application identifier pairs. The format of the GENTATTR_IDAPP request CT_IU is specified in table 538.

Table 538 – GENTATTR_IDAPP request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Control (0 r r r r r r r)	1
N_Port_ID #1	3
Application Identifier #1	see 6.9.3.3
...	
Control (1 r r r r r r r)	1
N_Port_ID #2	3
Application Identifier #n	see 6.9.3.3

One or more N_Port_ID field and Application Identifier pairs are specified. Each specified N_Port_ID field and Application Identifier pair is preceded by an 8-bit Control field. The format of the Control field is:

a) ~~Bit 7~~ is set to zero if the N_Port_ID field and Application Identifier pair following the Control field is not the last N_Port_ID field and Application Identifier pair to be specified by the request CT_IU. ~~Bit 7~~ is set to one if the N_Port_ID field and Application Identifier pair following the Control field is the last N_Port_ID field and Application Identifier pair specified by the accept CT_IU; and

b) Bits 6-0 are reserved.

If the GENTATTR_IDAPP request is not supported a reject CT_IU shall be returned with reason code ‘Command not supported’ and reason code explanation ‘No further explanation’.

The format of the accept CT_IU to a GENTATTR_IDAPP request is specified in table 539.

Table 539 – Accept CT_IU to GENTATTR_IDAPP request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Control (0 r r r r r r r)	1
N_Port_ID #1	3
Application Identifier Attribute object #1	see 6.9.3.3
...	
Control (1 r r r r r r r)	1
N_Port_ID #2	3
Application Identifier Attribute object #n	see 6.9.3.3

One or more Entity Attribute Allocation objects are returned. Each returned Entity Attribute Allocation object is preceded by an 8-bit Control field. The format of the Control field is:

a) ~~Bit 7~~ is set to zero if the Entity Attribute Allocation object following the Control field is not the last Entity Attribute Allocation object to be returned by the accept CT_IU; the bit is set to one if the Entity Attribute Allocation object following the Control field is the last Entity Attribute Allocation object returned by the accept CT_IU; and

b) Bits 6-0 are reserved.

If an Application Identifier specified in the request does not have an application identifier allocation, then no Entity Attribute Allocation object is returned for the specified Application Identifier.

The Application Server may return Entity Attribute Allocation objects in any order.

6.9.5.7 Query - Get All Entity Attributes (GALLENTATTR_ID)

The Application Server shall, if it receives a GALLENTATTR_ID request, return all the Entity Attribute Allocation objects for the specified N_Port_ID. The format of the GALLENTATTR_ID request CT_IU is specified in table 540

Table 540 – GALLENTATTR_ID request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Reserved	1
N_Port_ID	3

If the GALLENTATTR_ID request is not supported a reject CT_IU shall be returned with reason code ‘Command not supported’ and reason code explanation ‘No further explanation’.

The format of the accept CT_IU to a GALLENTATTR_ID request is specified in table 541.

Table 541 – Accept CT_IU to GALLENTATTR_ID request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Control (0 r r r r r r r)	1
Reserved	3
Entity Attribute Allocation object #1	see 6.9.3.3
...	
Control (1 r r r r r r r)	1
Reserved	3
Entity Attribute Allocation object #n	see 6.9.3.3

One or more Entity Attribute Allocation objects are returned. Each returned Entity Attribute Allocation object is preceded by an 8-bit Control field. The format of the Control field is:

a) ~~Bit 7~~ is set to zero if the Entity Attribute Allocation object following the Control field is not the last Entity Attribute Allocation object to be returned by the accept CT_IU; the bit is set to one if the Entity Attribute Allocation object following the Control field is the last Entity Attribute Allocation object returned by the accept CT_IU; and

b) Bits 6-0 are reserved.

6.9.5.8 Query - Get Entity Attributes (GENTATTR_ENT)

The Application Server shall, if it receives a GENTATTR_ENT request, return all the Entity Attribute Allocation objects in the Fabric for the specified entity identifiers. The format of the GENTATTR_ENT request CT_IU is specified in table 542.

Table 542 – GENTATTR_ENT request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Control (0 r r r r r r r)	1
Reserved	3
Entity Identifier object #1	see 6.9.3.2
...	
Control (1 r r r r r r r)	1
Reserved	3
Entity Identifier object #n	see 6.9.3.2

One or more Entity Identifier objects are specified. Each specified Entity Identifier object is preceded by an 8-bit Control field. The format of the Control field is:

a) ~~Bit 7~~ is set to zero if the Entity Identifier object following the Control field is not the last Entity Identifier object to be specified by the request CT_IU. ~~Bit 7~~ is set to one if the Entity Identifier object following the Control field is the last Entity Identifier object specified by the accept CT_IU; and

b) Bits 6-0 are reserved.

If the GENTATTR_ENT request is not supported a reject CT_IU shall be returned with reason code ‘Command not supported’ and reason code explanation ‘No further explanation’.

The format of the accept CT_IU to a GENTATTR_ENT request is specified in table 543.

Table 543 – Accept CT_IU to GENTATTR_ENT request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Control (0 r r r r r r r)	1
Reserved	3
Entity Attribute Allocation object #1	see 6.9.3.3
...	
Control (1 r r r r r r r)	1
Reserved	3
Entity Attribute Allocation object #n	see 6.9.3.3

One or more Entity Attribute Allocation objects are returned, and the number of Entity Attribute Allocation objects returned may be greater than the number of Entity Identifiers specified in the request. Each returned Entity Attribute Allocation object is preceded by an 8-bit Control field. The format of the Control field is:

a) ~~Bit 7~~ is set to zero if the Entity Attribute Allocation object following the Control field is not the last Entity Attribute Allocation object to be returned by the accept CT_IU; the bit is set to one if the Entity Attribute Allocation object following the Control field is the last Entity Attribute Allocation object returned by the accept CT_IU; and

b) Bits 6-0 are reserved.

If the entity identifier specified in the Entity Attribute Allocation object does not have an application identifier assignment, then the N_Port_ID field is set to all zeroes and the Application Identifier field value is set to ‘Unknown application identifier’ (see table 514) in the Application Identifier Allocation object field and the Number of Entity Attributes field is set to zero in the Entity Attribute Allocation object.

6.9.5.9 Query - Get All Entity Attributes (GALLENTATTR)

The Application Server shall, if it receives a GALLENTATTR request, return all the Entity Attribute Allocation objects. The format of the GALLENTATTR request CT_IU is specified in table 544.

Table 544 – GALLENTATTR request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

If the GALLENTATTR request is not supported a reject CT_IU shall be returned with reason code ‘Command not supported’ and reason code explanation ‘No further explanation’.

The format of the accept CT_IU to a GALLENTATTR request is specified in table 545.

Table 545 – Accept CT_IU to GALLENTATTR request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Control (0 r r r r r r r)	1
Reserved	3
Entity Attribute Allocation object #1	see 6.9.3.3
...	
Control (1 r r r r r r r)	1
Reserved	3
Entity Attribute Allocation object #n	see 6.9.3.3

One or more Entity Attribute Allocation objects are returned. Each returned Entity Attribute Allocation object is preceded by an 8-bit Control field. The format of the Control field is:

a) ~~Bit 7~~ is set to zero if the Entity Attribute Allocation object following the Control field is not the last Entity Attribute Allocation object to be returned by the accept CT_IU; the bit is set to one if the Entity Attribute Allocation object following the Control field is the last Entity Attribute Allocation object returned by the accept CT_IU; and

b) Bits 6-0 are reserved.

6.9.5.10 Register Application Identifier (RAPP_IDENT)

The RAPP_IDENT request shall be used to allocate application identifier(s) for a specified N_Port_ID and one or more entity identifier(s).

The Application Server shall not attempt validation of an entity identifier. This means that any Entity Identifier field value shall be accepted.

If the RAPP_IDENT request is not supported a reject CT_IU shall be returned with reason code ‘Command not supported’ and reason code explanation ‘No further explanation’.

The format of the RAPP_IDENT request CT_IU is specified in table 546.

Table 546 – RAPP_IDENT request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Reserved	1
N_Port_ID	3
Number of Entity Identifier objects	4
Entity Identifier object #1	see 6.9.3.2
Entity Identifier object #n	see 6.9.3.2

The format of the accept CT_IU to a RAPP_IDENT request is specified in table 547.

Table 547 – Accept CT_IU to RAPP_IDENT request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Control (0 r r r r r r r)	1
Reserved	3
Application Identifier Allocation object #1	see 6.9.3.1
...	
Control (1 r r r r r r r)	1
Reserved	3
Application Identifier Allocation object #n	see 6.9.3.1

One or more Application Identifier Allocation objects are returned. Each returned Application Identifier Allocation object is preceded by an 8 bit Control field. The format of the Control field is:

a) ~~Bit 7~~ is set to zero if the Application Identifier Allocation object following the Control field is not the last Application Identifier Allocation object to be returned by the accept CT_IU. Bit 7 is set to one if the Application Identifier Allocation object following the Control field is the last Application Identifier Allocation object returned by the accept CT_IU; and

b) Bits 6 to 0 are reserved.

If the Application Server is not able to allocate an application identifier for a specified entity identifier, then the Application Identifier field value in the Application Identifier Allocation object for the specified entity identifier shall be set to ‘Unknown application identifier’ (see table 524).

The Application Server may return Application Identifier Allocation objects in any order, and may return a list containing successful and unsuccessful application identifier allocations.

If the N_Port_ID in the RAPP_IDENT request does not match the N_Port_ID of the requestor, then a Reject CT_IU with reason code ‘Protocol error’ and reason code explanation ‘Access denied’ shall be returned.

6.9.5.11 Register Entity Attributes (RENTATTR_APPIA)

The RENTATTR_APPIA request shall be used to associate Entity attributes to a specified Application Identifier Allocation object.

The Application Server shall not attempt validation of an entity attribute. This means that any Entity Attribute field value shall be accepted.

If the RENTATTR_APPIA request is not supported a reject CT_IU shall be returned with reason code ‘Command not supported’ and reason code explanation ‘No further explanation’.

The format of the RENTATTR_APPIA request CT_IU is specified in table 548.

Table 548 – RENTATTR_APPIA request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Application Identifier Allocation object	see 6.9.3.1
Reserved	3
Number of Entity Attribute Entries	1
Entity Attribute Entry #1	see 6.9.3.4
Entity Attribute Entry #n	see 6.9.3.4

The format of the accept CT_IU to a RENTATTR_APPIA request is specified in table 549

Table 549 – Accept to RENTATTR_APPIA request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Entity Attribute Allocation object	see 6.9.3.3

One Entity Attribute Allocation object is returned.

If an Application Identifier Allocation object specified in the request is not found, then no Entity Attribute Allocation object is returned for the specified Application Identifier.

If the Application Server is unable to register any of the Entity Attributes specified, then the Number of Entity Attributes in the Entity Attribute Allocation object is set to zero (i.e. none of the Entity Attributes specified are registered).

If the N_Port_ID in the Application Identifier Allocation object does not match the N_Port_ID of the requestor, then a Reject CT_IU with reason code ‘Protocol error’ and reason code explanation ‘Access denied’ shall be returned.

6.9.5.12 Deregister Application Identifier (DAPP_IDENT)

The DAPP_IDENT request shall be used to deallocate an application identifier for a specified N_Port_ID and one or more entity identifier(s) and the Entity attributes associated with the Entity identifier(s).

If the DAPP_IDENT request is not supported a reject CT_IU shall be returned with a reason code of ‘Command not supported’ and reason code explanation of ‘No further explanation’.

The format of the DAPP_IDENT request CT_IU is shown in table 550.

Table 550 – DAPP_IDENT request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Reserved	1
N_Port_ID	3
Number of Entity Identifier objects	4
Entity Identifier object #1	see 6.9.3.2
Entity Identifier object #n	see 6.9.3.2

The format of the accept CT_IU to a DAPP_IDENT request is specified in table 551.

Table 551 – Accept CT_IU to DAPP_IDENT request

Item	Size (Bytes)
CT_IU preamble	see 4.3
Control (0 X X r r r r r)	1
N_Port_ID #1	3
Entity Identifier object #1	see 6.9.3.2
...	
Control (1 X X r r r r r)	1
N_Port_ID #n	3
Entity Identifier object #n	see 6.9.3.2

One or more N_Port_ID field and Entity Identifier object pairs are returned. Each returned N_Port_ID field and Entity Identifier object pair is preceded by an 8 bit Control field. The format of the Control field is:

a) Bit 7 is set to zero if the N_Port_ID field and Entity Identifier object pair following the Control field is not the last N_Port_ID field and Entity Identifier object pair to be returned by the accept CT_IU. Bit 7 is set to one if the N_Port_ID field and Entity Identifier object pair following the Control field is the last N_Port_ID field and Entity Identifier object pair returned by the accept CT_IU;

b) Bit 6 is set to one if the Application Server was not able to deallocate the application identifier for the specified N_Port_ID field and Entity Identifier object pair, and the deallocation should be retried. Bit 6 is set to zero to indicate the Application Server was able to deallocate the application identifier for the specified N_Port_ID field and Entity Identifier object pair;

c) Bit 5 is set to one if the Application Server does not have an application identifier allocation for the N_Port_ID field and Entity Identifier object pair. Bit 5 is set to zero if the Application Server had an Application Identifier Allocation object; and

d) Bits 4 to 0 are reserved.

The Application Server may return N_Port_ID field and Entity Identifier object pairs in any order.

If the N_Port_ID in the DAPP_IDENT request does not match the N_Port_ID of the requestor, then a Reject CT_IU with reason code ‘Protocol error’ and reason code explanation ‘Access denied’ shall be returned.

6.9.5.13 Deregister All Application Identifiers (DALLAPP_ID)

The DALLAPP_ID request shall be used to deallocate all Application Identifier Allocation objects and Entity attributes for the specified N_Port_ID.

If the DALLAPP_ID request is not supported a reject CT_IU shall be returned with reason code of ‘Command not supported’ and reason code explanation of ‘No further explanation’.

The format of the DALLAPP_ID request CT_IU is specified in table 552.

Table 552 – DALLAPP_ID request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Reserved	1
N_Port_ID	3

The format of the DALLAPP_ID accept CT_IU is specified in table 553.

Table 553 – DALLAPP_ID accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

If the N_Port_ID in the DALLAPP_ID request does not match the N_Port_ID of the requestor, then a Reject CT_IU with reason code ‘Protocol error’ and reason code explanation ‘Access denied’ shall be returned.

7 Event Service

7.1 Overview

The Event Service provides a registration and notification facility to notify Nx_Ports and Switches of events. The GS_Type for all Event Services shall be set as indicated in table 4. Table 554 defines the GS_Subtype codes for the Event Service.

Table 554 – Event Service Subtype Values

Values	Item
01h	Event Server
E0h to FFh	Vendor Specific Servers
other values	Registration Value

7.2 Event Server

7.2.1 Overview

The Event Server provides a way for Nx_Ports to register for and receive Event Notifications. Requests for the Event Server are carried over the Common Transport. The Event Server is intended to be distributed among Switches, making the Event Server immediately available to an Nx_Port once it has successfully completed Fabric Login. However, the Event Server is not restricted or required to be part of a Fabric, and may be located in any Nx_Port.

7.2.2 Protocol

7.2.2.1 Overview

Event Server registration, deregistration, and queries are managed through protocols containing a set of Request CT_IUs and Response CT_IUs supported by the Event Server.

For a Event Server request, the payload shall be transported from the requestor to the Event Server using a Request CT_IU. The corresponding Event Server response is transported from the Event Server to the requestor, in the Exchange established by the requestor, using a Response CT_IU. The action of the Event Server is unaffected by Server Sessions.

7.2.2.2 CT_IU Preamble Values

The following values shall be set in the CT_IU preamble for Event Server requests and their responses; fields not specified here shall be set as defined in 4.3.2:

a) GS_Subtype: as indicated in table 554; and

b) Command Code: See table 555 for Request command codes.

Table 555 – Event Server Request Command Codes

Code	Mnemonic	Description	Reference subclause
0100h	ER	Event Registration	7.2.3
0101h	EN	Event Notification	7.2.4
other	reserved		

7.2.3 Event Registration (ER)

7.2.3.1 Description

When the Event Server receives an ER, it shall register the Nx_Port for specific Event Notifications. Registrations are specific to Fibre Channel Services or general to the Fabric. ERs are sent as CT_IU in Class 3 to the Event Registration Server or to other Nx_Ports.

7.2.3.2 ER Request Payload

The ER Request Payload is sent to the Event Server or an Nx_Port that shall be required to send Event Notifications. The format of the ER request payload is shown in table 556.

Table 556 – ER Request CT_IU

Item	Size (Bytes)
CT_IU Preamble	see 4.3
Reserved	4
Number of Registrations (n)	4
Registration #1	see table 557
Registration #2	see table 557
...	
Registration #n	see table 557

Number of Registrations: The number of registrations that are included in this ER.

Registration: The format of the Registrations follows the Tag, Length, Value format of table 557.

Table 557 – Registration Format

Item	Size (Bytes)
Registration Tag	4
Registration Length	4
Registration Value	n

Registration Tag: The registration tags follow the format defined in table 558.

Table 558 – Registration Tag Format

Encoded Value	Grouping
00 xx xx xxh	General Events
FC xx xx xxh	Directory Server Events
FC 02 xx xxh	Name Server Events
FA xx xx xxh	Management Server Events
FA 01 xx xxh	Fabric Configuration Server Events
FF xx xx xxh	Vendor Specific Events
Other	Reserved

Registration Length: The length of the Registration Value in Bytes. The length shall be a multiple of four.

Registration Value: The value corresponding to each Registration Tag is defined in 7.2.3.3.

7.2.3.3 Registrations

7.2.3.3.1 Overview

Registration formats for each Registration Tag value are defined in this clause.

7.2.3.3.2 Name Server Registrations

Registration Tags: For the Name Server, the Registration Tags are defined in table 559.

Table 559 – Name Server Registration Tags

Encode Value	Description
FC 02 00 01h	Register for Nx_Port Events
FC 02 00 02h	Deregister for Nx_Port Events
FC 02 00 03h	Register for Name Server Updates
FC 02 00 04h	Deregister for Name Server Updates
Other Values	Reserved

Register for Nx_Port Events: To register for Event Notifications regarding Nx_Ports becoming available or unavailable, an Nx_Port shall register with this tag.

Deregister for Nx_Port Events: To deregister for Event Notifications regarding Nx_Ports becoming available or unavailable, an Nx_Port shall register with this tag.

Register for Name Server Updates: To register for Event Notifications regarding changes to Nx_Ports Name Server objects, an Nx_Port shall register with this tag. The Name Server Updates are for Name Server registrations that occur after the initial Nx_Port Available Event Notification was sent. The Nx_Port registering for this event shall also be registered for Nx_Port Events.

Deregister for Name Server Updates: To deregister for Event Notifications regarding changes to Nx_Ports Name Server objects, an Nx_Port shall register with this tag.

Registration Length: The length of each registration is 0 Bytes.

Registration Value: The value of each Name Server Registration is NULL.

7.2.3.3.3 Management Server Registrations

For the Management Server, the Registration Tags are defined in table 560.

Table 560 – Management Server Registration Tags

Encode Value	Description
FA 00 00 01h	Link Incident Record Registrations
Other Values	Reserved

Link Incident Record Registrations: To register for Registered Link Incident Records via Event Registrations, an Nx_Port shall embed the LIRR payload in the Event Registration. The format for LIRR Registrations are defined in table 561.

Table 561 – Register for Registered Link Incident Records

Bits Word	31 to 24	23 to 16	15 to 08	07 to 00
0	Embedded LIRR Payload			
1				

Embedded LIRR Payload: The payload for Link Incident Record Registration is defined in FC-LS-3. The payload includes the LIRR command code.

7.2.3.4 Accept CT_IU for ER Request

Each acceptance or rejection of an ER shall not depend on the acceptance or rejections of individual Registrations within the ER. When a registration is accepted, the ER Accept CT_IU shall not contain that registration in the rejected registrations. If the registration is not successful, the registration shall be returned as a rejected registration. The ER Accept CT_IU payload is defined in table 562.

Table 562 – Accept CT_IU to ER

Item	Size (Bytes)
CT_IU Preamble	See 4.3
Number of Accepted Registrations	4
Number of Rejected Registrations (n)	4
Registration #1	See table 557
Reason for Rejected Registration	4
Registration #2	See table 557
Reason for Rejected Registration	4
...	
Registration #n	See table 557
Reason for Rejected Registration	4

Number of Accepted Registrations: The number of registrations that were accepted in the ER request.

Number of Rejected Registrations: The number of registrations that were rejected in the ER request.

Registration: The format of the Registration is defined in table 557. The Rejected Registration shall include an exact copy of the registration that failed.

Reason for Rejected Registration: The reason for the rejected registration is defined in table 563. If a registration is rejected, the replying Switch shall register all readable fields. Duplicate tags should be accepted, but the last field shall be registered.

Table 563 – Reasons For Rejected Registrations

Item	Value
Registration Tag Not Supported	00 00 00 01h
Invalid Registration Value	00 00 00 02h
Registration Full	00 00 00 03h
Duplicate Registration	00 00 00 04h
Invalid Object Length	00 00 00 05h
Others	Reserved

7.2.4 Event Notification (EN)

7.2.4.1 Description

An EN is sent from the Event Server to an Nx_Port or another Fabric Entity after an event occurs. Events are specific to basic Fibre Channel Services or general to the Fabric. EN may include information about the event that is being reported so that devices do not have to query the Fabric to understand the event. Event Notification requests are CT_IUs sent in Class 3.

7.2.4.2 Event Notification Request CT_IU

The format of the EN request CT_IU is shown in table 564.

Table 564 – Event Notification Request CT_IU

Item	Size (Bytes)
CT_IU Preamble	see 4.3
Number of Events (n)	4
Event #1	see table 565
Event #2	see table 565
...	
Event #n	see table 565

Number of Events: The number of events that are included in this EN.

Event: The format of the Events follows the Tag, Length, Value format of table 565.

Table 565 – Event Format

Item	Size (Bytes)
Event Tag	4
Event Length	4
Event Value	n

Event Tag: The event tags follow the format defined in table 558.

Event Length: The length of the Event Value in Bytes. The length shall be a multiple of four.

Event Value: The value corresponding to each Event Tag is specific to each grouping and is defined in 7.2.4.3.

7.2.4.3 Events

7.2.4.3.1 Overview

Event Values for each Event Tag are defined in this clause.

7.2.4.3.2 Name Server Event Notifications

Event Tags: For the Name Server, the Event Tags are defined in table 566.

Table 566 – Name Server Event Tags

Encode Value	Description
FC 02 00 01h	Nx_Port Available
FC 02 00 02h	Nx_Port Unavailable
FC 02 00 03h	Name Server Updated
Other Values	Reserved

Event Length: The length of the Event Value in Bytes.

Event Values: The Name Server Event Values are defined in table 567. The values for each Name Server Event tag are defined in this clause.

Nx_Port Available: This Event notifies the Nx_Port that the specified Nx_Port is available for communication. Port Entries shall contain all registered Name Server Objects for the N_Port_ID.

Nx_Port Unavailable: This Event notifies the Nx_Port that the specified Nx_Port is unavailable for communication. Port Entries shall include the N_Port_ID and Port_Name Object for the unavailable Nx_Port.

Name Server Updated: This Event notifies the Nx_Port that the Name Server Objects for the specified Nx_Port have changed. The Port Entry shall contain all registered Name Server Objects for each updated Nx_Port.

Table 567 – Name Server Event Format

Item	Size (Bytes)
Number of Port entries (r)	4
Port Entry # 1	see table 568
Port Entry # 2	see table 568
...	
Port Entry # r	see table 568

Number of Port Entries: The number of Port Entries in this Event.

Port Entry: The format of the Port Entry is defined in table 568.

Table 568 – Port Entry Format

Item	Size (Bytes)
N_Port_ID	3
Reserved	1
Record Length	4
Name Server Object #1	x
Name Server Object #2	y
...	...
Name Server Object #n	z

N_Port_ID: The N_Port_ID that is associated with the Name Server Objects.

Record Length: The number of bytes in the Name Server Objects.

Name Server Object: The format of the Name Server Objects are defined in table 569. The order of the Name Server Objects within a Port Entry may be in any order.

Table 569 – Name Server Object Format

Item	Size (Bytes)
Name Server Object Tag	2
Name Server Object Length	2
Name Server Object Value	n

Name Server Object Tag: The Name Server Object tag is defined in table 570.

Table 570 – Name Server Object Tag

Encoded Value	Description
00 01h	Port Type
00 02h	N_Port_Name
00 03h	Port Symbolic Name
00 04h	Node Name
00 05h	Node Symbolic Name
00 06h	Obsolete
00 07h	Obsolete
00 08h	Class of Service
00 09h	FC-4 Types
00 0Ah	Obsolete
00 0Bh	F_Port_Name
00 0Ch	Hard Address
00 0Dh	FC-4 Features
00 0Eh	Obsolete
00 0Fh	Permanent Port Name
00 10h	Core Port Name
00 11h	Small Name Server Entry Object (see FC-SW-6)
00 F0h to 00 FFh	Vendor Specific
Others	Reserved

Name Server Object Length: The Name Server Object Length is the length of the following Name Server Object Value plus fill bytes. Fill bytes are added as necessary to the end of the actual value in order to ensure that the length of the value is a multiple of four. Fill bytes shall be 00h.

Name Server Object Value: The value of the Name Server Object identified by the Name Server Object Tag plus fill bytes. The Name Server Objects are defined in 5.2.3. Vendor Specific Values shall include the T10 Vendor ID (see 3.6) as the first 8 bytes of each entry.

7.2.4.3.3 Management Server Notifications

For the Management Server, the Event Tags are defined in table 571.

Table 571 – Management Server Values

Encode Value	Description
FA 00 00 01h	Registered Link Incident Records
Other Values	Reserved

Registered Link Incident Records: This Event Notification embeds the RLIR payload into an Event Notification. The format for the notification is defined in table 572.

Table 572 – Registered Link Incident Record Notification

Bits Word	31 to 24	23 to 16	15 to 08	07 to 00
...	Embedded RLIR Payload			

Embedded RLIR Payload: The RLIR Payload defined in FC-LS-3 shall be embedded in the Event Value.

7.2.4.4 Event Notification Accept CT_IU

The EN Accept CT_IU signifies acceptance of at least part of the EN Request. Each EN is accepted or rejected depending on if the receiving device supports the notification. When a notification is accepted, the EN Accept CT_IU shall not contain that event in the rejected notifications. If the notification is not successful, the notification shall be returned as a rejected notification. The EN CT_IU Accept payload is defined in table 573.

Table 573 – Accept CT_IU for EN

Item	Size (Bytes)
CT_IU Preamble	see 4.3
Number of Accepted Notifications	4
Number of Rejected Notifications (n)	4
Notification #1	see table 565
Reason for Rejected Notification	4
Notification #2	see table 565
Reason for Rejected Notification	4
...	
Notification #n	see table 565
Reason for Rejected Notification	4

Number of Accepted Notifications: The number of notifications that were accepted in the EN request.

Number of Rejected Notifications: The number of notifications that were rejected in the EN request.

Notification Format: The format of the Notification Entries is defined in table 565. The Rejected Notifications shall include an exact copy of the notification that failed.

Reason for Rejected Notification: The reason for the rejected Notification is defined in table 574.

Table 574 – Reasons for Rejected Notifications

Item	Value
Notification Tag Not Supported	00 00 00 01h
Invalid Notification Value	00 00 00 02h
Others	Reserved

Annex A: FC-4 Feature Bits (normative)

A.1 Overview

This clause documents proposed FC-4 specific definitions for the FC-4 Feature object that may be retrieved using the GFF_ID and GID_FF Name Server queries (see 5.2).

A.2 FC-4 Feature Bits

The Generic Fibre Channel Features FC-4 (type = DEh) shall use the FC-4 Feature Bits as follows:

Table A.1 – Generic Fibre Channel Features definition of FC-4 Feature Bits

Bit	Interpretation
3	Reserved
2	Reserved
1	Reserved
0	1 = Simplified Behavior declared (see FC-SCM) 0 = Behavior not declared

Annex B: Enhanced Fabric Configuration Server Discovery Examples (Informative)

B.1 Introduction

The purpose of this annex is to describe the general mechanisms and protocols that may be used to discover the components that comprise a Fibre Channel storage area network (SAN) using the ~~Enhance~~ Configuration Server.

B.2 Basic Assumptions

The use cases are based on the following basic assumptions:

a) The Fibre Channel SAN environment is viewed from a top down system perspective. Rather than being host, Fabric, or storage centric, the management capabilities and topology are viewed so that management applications may reside in the most appropriate location for a given management discipline;

b) From a management perspective, a management application may view the entire Fibre Channel SAN. This is different from the operational perspective, wherein certain aspects and views of the SAN may be restricted;

c) The Fibre Channel SAN's discovery and management mechanisms should allow for a variety of management implementations. For example, configuration and management information should be identified independent of the access mechanism that may be either in-band or out-of-band; and

d) This annex is intended to augment, but not necessarily replace existing discovery proposals.

B.3 Definitions

B.3.1 Fabric

As defined in FC-SW-6, an entity that interconnects various Nx_Ports attached to it, and is capable of routing frames using only the D_ID information in a FC-2 Frame_Header.

B.3.2 Management application

An abstract entity that communicates with management access points and Servers to provide management functions on behalf of a Fibre Channel SAN.

B.4 Simple Fibre Channel Fabric

B.4.1 Overview

The simple fibre channel example consists of a single attached port for an initiator and a single attached port for a target connected by an interconnect element. Figure 28 below depicts this view.

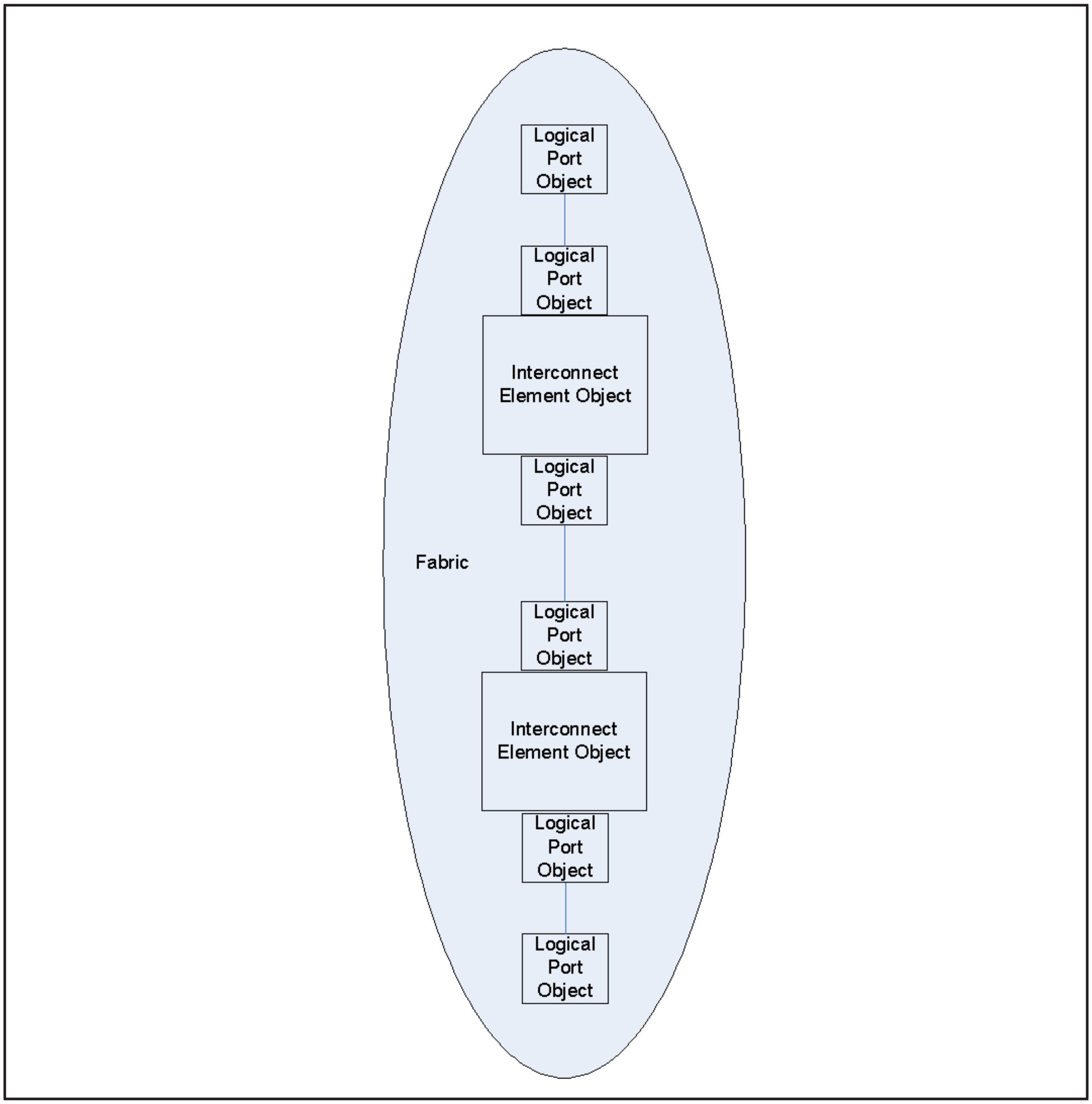


Figure 28 – Enhanced Fabric Configuration Server Objects Logical Model

B.4.2 Procedure

Use eGIEL to get the Interconnect Element Object Name attribute.

With the Name attribute, use eGIEAB to get each Interconnect Element Attribute Block. Each Interconnect Element's Transport Infrastructure Attribute should have a Transport Infrastructure Object Attribute Entry Type of Physical Object Correlatable Identifier and a Physical Object

Correlatable Identifier. With the Physical Object Correlatable Identifier, use eGPOAB to get each Interconnect Element's Physical Object Attribute Block.

With the Name attribute, use eGIEPL to get the two Logical Port's Object Names for each interconnect element. With the Name attribute, use eGLPAB to get each ports Logical Port Attribute Block and use eGAPL to get the Logical Port Object Name of the attached port.

Each Logical Port Attribute Block should have a Physical Port Correlatable Identifier. With the Physical Port Correlatable Identifier, use eGPPAB to get each Physical Port Attribute Block.

B.5 Fibre Channel Fabric with FCF and FDF Discovery

B.5.1 Overview

The fibre channel example with Controlling Switches and FCDFs consists of a switch connected to a Controlling Switch that is connected to an FCDF. Figure 29 below depicts this view.

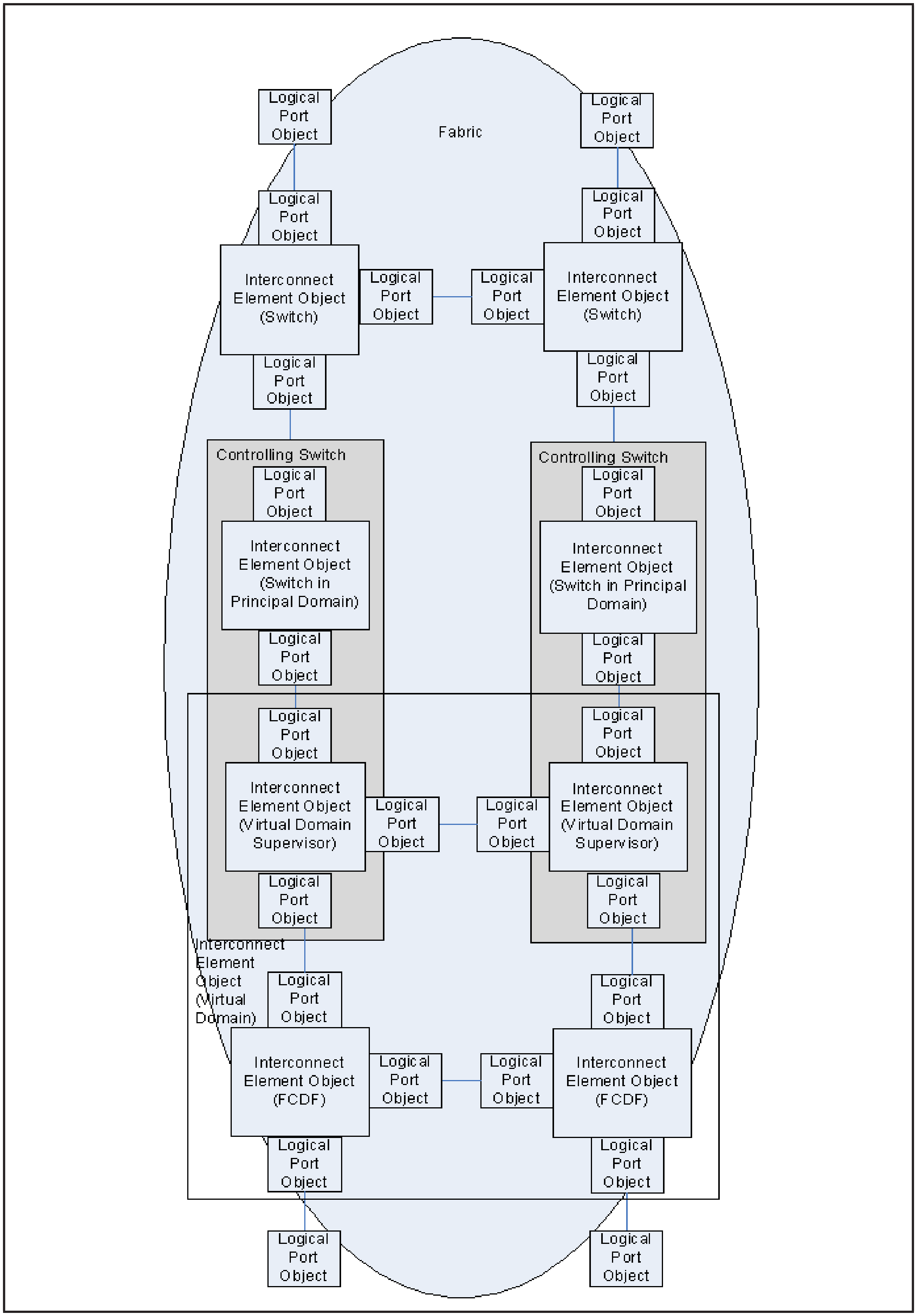


Figure 29 – Fibre Channel Fabric with Virtual Domain Supervisor and FCDF

B.5.2 Procedure

Use eGIEL to get the Interconnect Element Object Name attribute. There will be five Interconnect Element Objects returned. They are:

1) Two interconnect elements that are not Controlling Switches or FCDF identified as a switch.

2) Two interconnect element of the Controlling Switch that presents the principal domain.

3) The Interconnect element of the virtual domain.

With the Name attribute, use eGIEAB to get each Interconnect Element Attribute Block.

For the two interconnect elements identified as a switch, the Interconnect Element's Transport Infrastructure Attribute should have a Transport Infrastructure Object Attribute Entry Type of Physical Object Correlatable Identifier and a Physical Object Correlatable Identifier. With the Physical Object Correlatable Identifier, use eGPOAB to get each Interconnect Element's Physical Object Attribute Block.

For the two interconnect elements of the Controlling Switch that presents the principal domain, with the Name attribute, use eGIEAB to get each Interconnect Element Attribute Block. Each Interconnect Element's Transport Infrastructure Attribute should have a Transport Infrastructure Object Attribute Entry Type of Interconnect Element Object Name and an Interconnect Element Object Name.

Correlatable Identifier and a Physical Object Correlatable Identifier. With the Physical Object Correlatable Identifier, use eGPOAB to get each Interconnect Element's Physical Object Attribute Block.

With the Name attribute, use eGIEPL to get the two Logical Ports Object Names for each interconnect element. With the Name attribute, use eGLPAB to get each ports Logical Port Attribute Block and use eGAPL to get the Logical Port Object Name of the attached port.

Each Logical Port Attribute Block should have a Physical Port Correlatable Identifier. With the Physical Port Correlatable Identifier, use eGPPAB to get each Physical Port Attribute Block.