INTERNATIONAL STANDARD

ISO/IEC 14165-414

First edition
2007-05

Information technology – Fibre channel –

Part 414: Generic services-4 (FC-GS-4)

Reference number
ISO/IEC 14165-414:2007(E)

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland
Email: inmail@iec.ch
Web: www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

- Catalogue of IEC publications: www.iec.ch/searchpub

The IEC on-line Catalogue enables you to search by a variety of criteria (reference number, text, technical committee,…). It also gives information on projects, withdrawn and replaced publications.

- IEC Just Published: www.iec.ch/online_news/justpub

Stay up to date on all new IEC publications. Just Published details twice a month all new publications released. Available on-line and also by email.

- Customer Service Centre: www.iec.ch/webstore/custserv

If you wish to give us your feedback on this publication or need further assistance, please visit the Customer Service Centre FAQ or contact us:

Email: csc@iec.ch
Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00

INTERNATIONAL STANDARD

ISO/IEC 14165-414

First edition
2007-05

Information technology –
Fibre channel –

Part 414:
Generic services-4 (FC-GS-4)

PRICE CODE XH

For price, see current catalogue

Contents

INFORMATION TECHNOLOGY –
FIBRE CHANNEL –

Part 414: Generic services-4 (FC-GS-4)

FOREWORD

1) ISO (International Organization for Standardization) and IEC (International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards. Their preparation is entrusted to technical committees; any ISO and IEC member body interested in the subject dealt with may participate in this preparatory work. International governmental and non-governmental organizations liaising with ISO and IEC also participate in this preparation.

2) In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

3) The formal decisions or agreements of IEC and ISO on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC and ISO member bodies.

4) IEC, ISO and ISO/IEC Publications have the form of recommendations for international use and are accepted by IEC and ISO member bodies in that sense. While all reasonable efforts are made to ensure that the technical content of IEC, ISO and ISO/IEC Publications is accurate, IEC or ISO cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.

5) In order to promote international uniformity, IEC and ISO member bodies undertake to apply IEC, ISO and ISO/IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any ISO/IEC Publication and the corresponding national or regional publication should be clearly indicated in the latter.

6) ISO and IEC provide no marking procedure to indicate their approval and cannot be rendered responsible for any equipment declared to be in conformity with an ISO/IEC Publication.

7) All users should ensure that they have the latest edition of this publication.

8) No liability shall attach to IEC or ISO or its directors, employees, servants or agents including individual experts and members of their technical committees and IEC or ISO member bodies for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication of, use of, or reliance upon, this ISO/IEC publication or any other IEC, ISO or ISO/IEC publications.

9) Attention is drawn to the normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

10) Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

International Standard ISO/IEC 14165-414 was prepared by subcommittee 25: Interconnection of information technology equipment, of ISO/IEC joint technical committee 1: Information technology.

The list of all currently available parts of the ISO/IEC 14165 series, under the general title *Information technology – Fibre Channel,* can be found on the IEC web site.

This International Standard has been approved by vote of the member bodies and the voting results may be obtained from the address given on the second title page.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

INTRODUCTION

FC-GS-4 is one of the Fibre Channel family of standards. This family includes FC-FS, which specifies the Framing and Signalling Interface. FC-SW-2 is related to Fabric requirements. FC-AL-2 specifies the arbitrated loop topology.

FC-GS-4 describes in detail the basic Fibre Channel services introduced in FC-FS. In addition, this document describes any ancillary functions and services required to support the Fibre Channel services. Services described include name services, management services, discovery services, time services and alias services.

FC-GS-4 provides generic services that may be utilized by any upper layer protocol that makes use of Fibre Channel as a transport.

Figure 0 shows the relationship of this FC-GS-4 standard (highlighted in red) with other Fibre Channel standards and draft proposed standards.

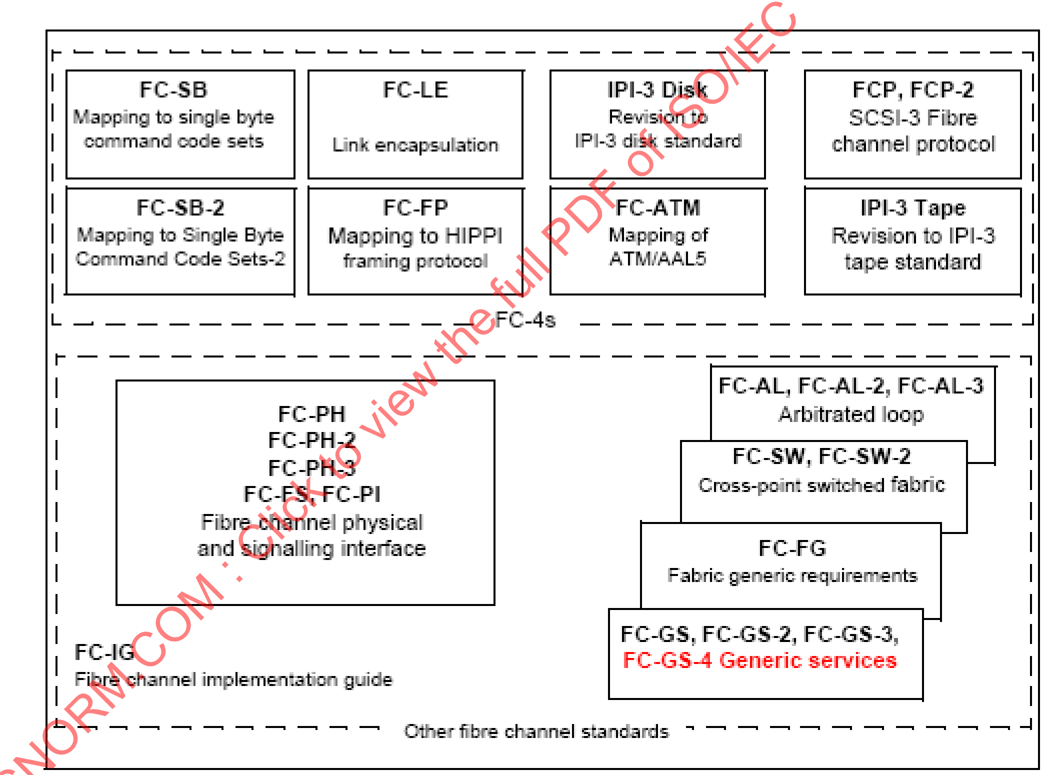

FC-SB
Mapping to single byte command code sets
FC-LE
Link encapsulation
IPI-3 Disk
Revision to IPI-3 disk standard
FCP, FCP-2
SCSI-3 Fibre channel protocol
FC-SB-2
Mapping to Single Byte Command Code Sets-2
FC-FP
Mapping to HIPPI framing protocol
FC-ATM
Mapping of ATM/AAL5
IPI-3 Tape
Revision to IPI-3 tape standard
FC-4s
FC-PH
FC-PH-2
FC-PH-3
FC-FS, FC-PI
Fibre channel physical and signalling interface
FC-AL, FC-AL-2, FC-AL-3
Arbitrated loop
FC-SW, FC-SW-2
Cross-point switched fabric
FC-FG
Fabric generic requirements
FC-GS, FC-GS-2, FC-GS-3,
FC-GS-4 Generic services
FC-IG
Fibre channel implementation guide
Other fibre channel standards

Figure 0 – Document relationship

The information presented in this document is grouped into clauses:

– Clause 1, Scope

- Clause 2, Normative references
- Clause 3, Definitions and conventions
- Clause 4, Common transport for generic services
- Clause 5, Directory service
- Clause 6, Management service
- Clause 7, Alias service
- Clause 8, Key distribution service
- Annex A, Service interface provided by FC-CT
- Annex B, FC-4 Feature bits
- Annex C, Discovery
- Annex D, Example of zone services
- Annex E, Time service
- Annex F, Performance server

INFORMATION TECHNOLOGY –
FIBRE CHANNEL –

Part 414: Generic services-4 (FC-GS-4)

1 Scope

FC-GS-4 describes in detail the basic Fibre Channel services introduced in FC-FS.

The Fibre Channel services described in this document are

- Directory Service,
- Management Service and
- Alias Service.

In addition to the aforementioned Fibre Channel services, the Common Transport (CT) protocol is described. The Common Transport service provides a common FC-4 for use by the Fibre Channel services.

2 Normative references

2.1 Overview

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

2.2 Approved references

ISO/IEC 14165-122, *Information technology – Fibre channel – Part 122: Arbitrated loop (FC-AL-2)* [ANSI INCITS 332-1999]

ISO/IEC 14165-251, *Information technology – Fibre channel – Part 251: Framing and signaling (FC-FS)* [ANSI INCITS 373-2003]

ISO/IEC 14776-222, *Information technology – Small computer system interface (SCSI) – Part 222: Fibre channel protocol for SCSI, 2nd Version (FCP-2)* [ANSI INCITS 350-2003].

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.

ISO/IEC 14165-133, *Information technology – Fibre channel – Part 133*: *Switch Fabric-3 (FC-SW-3)* [ANSI INCITS 384-2004]

ISO/IEC 14165-251, *Information technology – Fibre channel – Part 251*: *Primary commands-3 (SPC-3)* [ANSI INCITS 408-2005]

ISO/IEC 14776-412, *Information technology – Small computer system interface (SCSI) – Part 412*: *Architecture model-2 (SAM-2)*

ISO/IEC 14776-413, *Information technology – Small computer system interface (SCSI) – Part 413*: *Architecture model - 3 (SAM-3)* [ANSI INCITS 402-2005]

2.4 Other references

All references in this subclause were correct at the time of approval of this International Standard. The provisions of the referenced specifications, as identified in this subclause, are valid within the context of this International Standard. The reference to a specification within this International Standard does not give it any further statuswithin ISO/IEC; in particular, it does not give the referenced specification the status of an International Standard.

The following RFC documents are available from the RFC Editor, Information Sciences Institute, University of Southern California, 4676 Admiralty Way, Suite 1001, Marina del Rey, CA 90292-6695; (310) 822-1511 or (310) 823-6714 (fax); http://www.isi.edu/.

RFC 791, *Internet Protocol, STD 5*

RFC 1305, *Network Time Protocol (Version 3) Specification, Implementation and Analysis*, March 1992.

RFC 1321, *The MD5 Message-Digest Algorithm*, April 1992.

RFC 2030, *Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI*, October 1996.

RFC 2104, *HMAC: Keyed-Hashing for Message Authentication*, February 1997.

RFC 2373*, IP Version 6 Addressing Architecture*, July 1998.

ANSI X9.30.2-1997, *Public Key Cryptography for the Financial Services Industry - Part 2: The Secure Hash Algorithm (SHA-1).*

ANSI INCITS 386-2004, *Information technology – Fibre Channel – HBA API (FC-HBA)*, Rev 13

INCITS TR-30-2002, *Information technology – Fibre Channel – Methodologies for Interconnects (FC-MI)*.

3 Definitions and conventions

3.1 Overview

For FC-GS-4, the following definitions, conventions, abbreviations, acronyms and symbols apply.

3.2 Definitions

3.2.1
address identifier
an address value used to identify source (S_ID) or destination (D_ID) of a frame

3.2.2
alias address identifier (alias)
one or more address identifiers that may be recognized by an Nx_Port in addition to its N_Port Identifier;
an alias address identifier is Fabric unique and may be common to multiple Nx_Ports

3.2.3
Area_ID:
second level in a three-level addressing hierarchy (see FC-SW-3)

3.2.4
B_Port
a Bridge Port is a Fabric inter-element port used to connect Bridge devices with E_Ports on a Switch;
the B_Port provides a subset of the E_port functionality (see FC-SW-3)

3.2.5
bridge
see FC-FS

3.2.6
broadcast zone
zone with the broadcast attribute set

3.2.7
broadcast zoning enforcement
zoning technique in which the fabric limits the broadcast distribution among zone members through frame-by-frame filtering techniques

3.2.8
client
entity that, via its CT, makes requests of a server

3.2.9
common transport (CT)
a protocol defined by this standard that provides access to Services and their related servers;
CT may also refer to an instance of the common transport

3.2.10
default zone
a zone that contains all Nx_Ports that are not members of any zone in the active zone set;
the default zone is not a member of the active zone set

3.2.11
directory
repository of information about objects that may be accessed via the directory service

3.2.12
Domain_ID
the highest or most significant hierarchical level in the three-level addressing hierarchy (i.e., the most significant byte of the address identifier) (see FC-SW-3)

3.2.13
E_Port
a fabric “expansion” port that attaches to another Interconnect_Port to create an inter-switch link (see FC-SW-3)

3.2.14
F_Port
the link control facility within the fabric that attaches to an N_Port through a link (see FC-FS)

3.2.15
FL_Port
an F_Port that contains arbitrated loop functions associated with arbitrated loop topology (see FC-AL-2).

3.2.16
Fx_Port
a switch port capable of operating as an F_Port or FL_Port

3.2.17
fabric
any interconnect between two or more Nx_Ports, including point-to-point, loop and switched fabric

3.2.18
FC_Port
port that is capable of transmitting or receiving fibre channel frames (see FC-FS)

3.2.19
F/NL_Port
NL_Port that detects OPN(00,x) and provides fibre channel services in the absence of an FL_Port (see FC-AL-2)

3.2.20
gateway
see FC-FS

3.2.21
generic services
collection of services defined by this standard

3.2.22
hard zone
zone with the hard zoning attribute set

3.2.23
hard zoning enforcement

zoning technique in which the fabric limits frame exchange by frame-by-frame filtering

3.2.24
host
see FC-FS

3.2.25
host bus adapter
I/O adapter that connects a host computer bus to the fibre channel medium

3.2.26
hub
see FC-FS

3.2.27
interconnect element
any device in a fabric that assists in the transport of fibre channel frames between FC_Ports

3.2.28
link control facility
a hardware facility that attaches to an end of a link and manages transmission and reception of data; it is contained within each FC_Port (see FC-FS)

3.2.29
link service facilitator
the entity at WKA identifier FFFFFEh, known as the fabric F_Port in FC-FS, the address identifier to which fabric login is directed

3.2.30
multi-function device
see FC-FS

3.2.31
NAS server
see FC-FS

3.2.32
N_Port
a hardware entity that includes a link control facility;
it may act as an Originator, a Responder, or both

3.2.33
NL_Port
an L_Port that is able to perform the function of an N_Port, attached via a link to one or more NL_Ports and zero or more FL_Ports in an arbitrated loop topology (see FC-AL-2)

3.2.34
Nx_Port
a Port operating as an N_Port or NL_Port

3.2.35
N_Port identifier
a fabric unique address identifier by which an Nx_Port is uniquely known;
the identifier may be assigned by the fabric during the initialization procedure;

the identifier may also be assigned by other procedures not defined in FC-FS

3.2.36
N_Port_Name
as defined in FC-FS

3.2.37
Name_Identifier
a 64 bit identifier, with a 60 bit value preceded with a four bit Network_Address_Authority_Identifier, used to identify physical entities in fibre channel (for example, N_Port, Node, F_Port, or Fabric)

3.2.38
Network_Address_Authority (NAA)
an organization such as ITU or IEEE that administers network addressee

3.2.39
Network_Address_Authority identifier
A four bit identifier that indicates a Network_Address_Authority (NAA) (see FC-FS).

3.2.40
printable ASCII characters
ASCII characters in the range 20h through 7Eh

3.2.41
Requesting_CT
a CT that is sending a request

3.2.42
Responding_CT
a CT that is responding to a prior request

3.2.43
secured association identifier (SAID)
a value that specifies a secured association relationship between two clients;
SAIDs allow multiple concurrent secured associations to be active between two clients

3.2.44
server
a server is an entity that, via its CT, accepts requests from a client and provides responses to the client;
a server is accessed via a service (for example, the name server is accessed using the directory service)

3.2.45
service
a service is provided by a node, accessible via an N_Port that is addressed by a WKA or an N_Port identifier (for example, the directory service and the alias service);
a service provides access to one or more servers

3.2.46
soft zone
a zone consisting of zone members that are made visible to each other through client service requests;
typically, soft zones contain zone members that are visible to devices via name server exposure of

zone members;
the fabric does not enforce a soft zone on a frame-by-frame basis

3.2.47
soft zoning enforcement
zoning technique in which the fabric enforces membership through name server visibility

3.2.48
storage access device
see FC-FS

3.2.49
storage device
see FC-FS

3.2.50
storage subsystem
see FC-FS

3.2.51
switch
a fabric element conforming to the fibre channel switch fabric set of standards (see FC-SW-3)

3.2.52
switched fabric
a fabric comprised of one or more switches (see FC-SW-3)

3.2.53
symbolic name
a user-defined name for an object, up to 255 characters in length;
the directory service does not guarantee uniqueness of its value

3.2.54
unidentified Nx_Port
an Nx_Port that has not yet had its N_Port identifier assigned by the initialization procedure

3.2.55
virtualization device
see FC-FS

3.2.56
wavelength division multiplexer
see FC-FS

3.2.57
well-known address
an address identifier defined in FC-FS to access a service
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)

3.2.58
zone
a group of zone members; members of a zone are made aware of each other, but not made aware of Nx_Ports outside the zone

3.2.59
zone definition
the parameters that define a zone (i.e., the zone name, number of zone members and zone member definition)

3.2.60
zone member
an Nx_Port to be included in a zone, as specified by its zone member definition:
N_Ports at WKAs should not be specified as zone members

3.2.61
zone member definition
parameter by which a zone member is specified

3.2.62
zone name
name assigned to a zone

3.2.63
zone set
one or more zones that may be activated or deactivated as a group

3.2.64
zone set name
the name assigned to a zone set

3.2.65
zone set state
state of a zone set that is either activated or deactivated

3.2.66
active zone set
the zone set that is currently active:
only one zone set may be active at any time

3.3 Editorial conventions

In FC-GS-4, 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 (for example, Exchange, Class). Any lowercase uses of these words have the normal technical English meanings.

Lists sequenced by letters (e.g., a-red, b-blue, c-green) show no priority relationship between the listed items. Numbered lists (e.g., 1-red, 2-blue, 3-green) show a priority ordering between the listed items.

The ISO/IEC convention of numbering is used (i.e., the thousands and higher multiples are separated by a space and a comma is used as the decimal point). A comparison of the American and ISO conventions are shown in table 1.

Table 1 – ISO/IEC and American conventions

ISO/IEC	American
0,6	0.6
1 000	1,000
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 clauses.

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 clauses.

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.

CT	Common Transport
CT_IU	Common Transport Information Unit
D_ID	Destination address identifier
ELS	Extended Link Service
GBIC	Gigabit Interface Converter
GLM	Gigabit Link Module
GS	Fibre Channel Generic Service
HBA	Host Bus Adapter
IN_ID	Initial Identifier
IP	Internet Protocol
IPA	Initial Process Associator
IU	Information Unit
LUID	Logical Unit Unique Identifier (see FC-HBA)
LUN	Logical Unit Number (see SAM-3)

NAA	Network Address Authority
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 (see Annex E)
UDP	User Datagram Protocol
ULP	Upper Level Protocols
WKA	Well-Known Address

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.

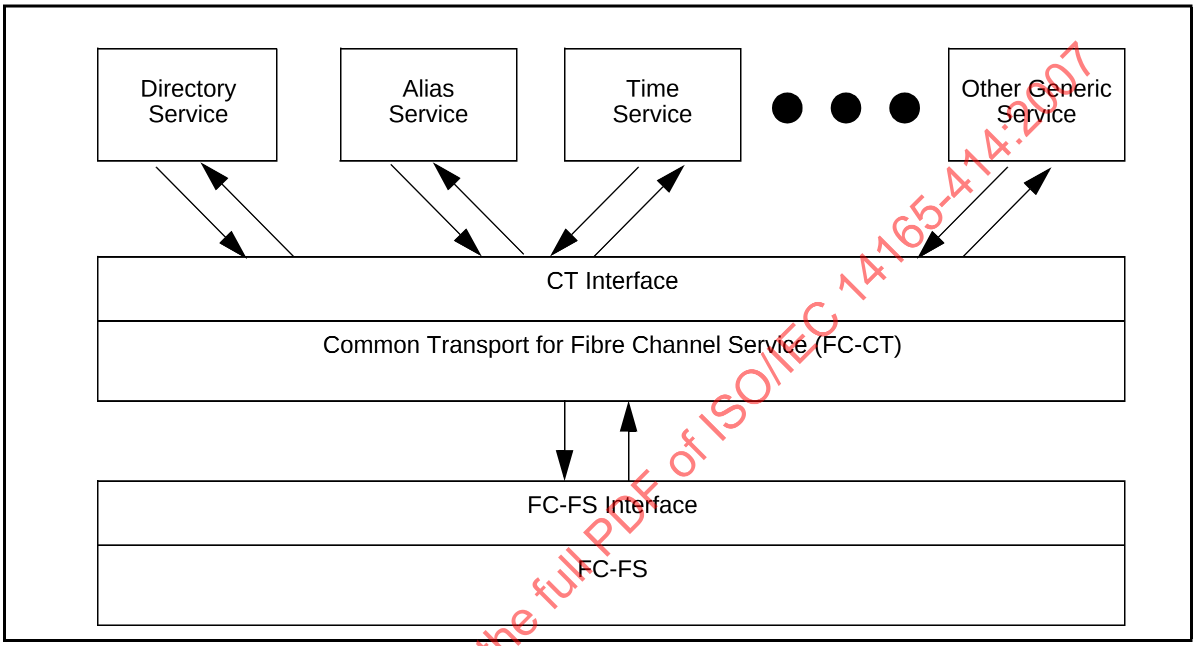


Figure 1 – Relationship of the Common Transport with Generic Services and FC-FS

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 WKA specific to the Service or at any N_Port Identifier where it is offered. 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.

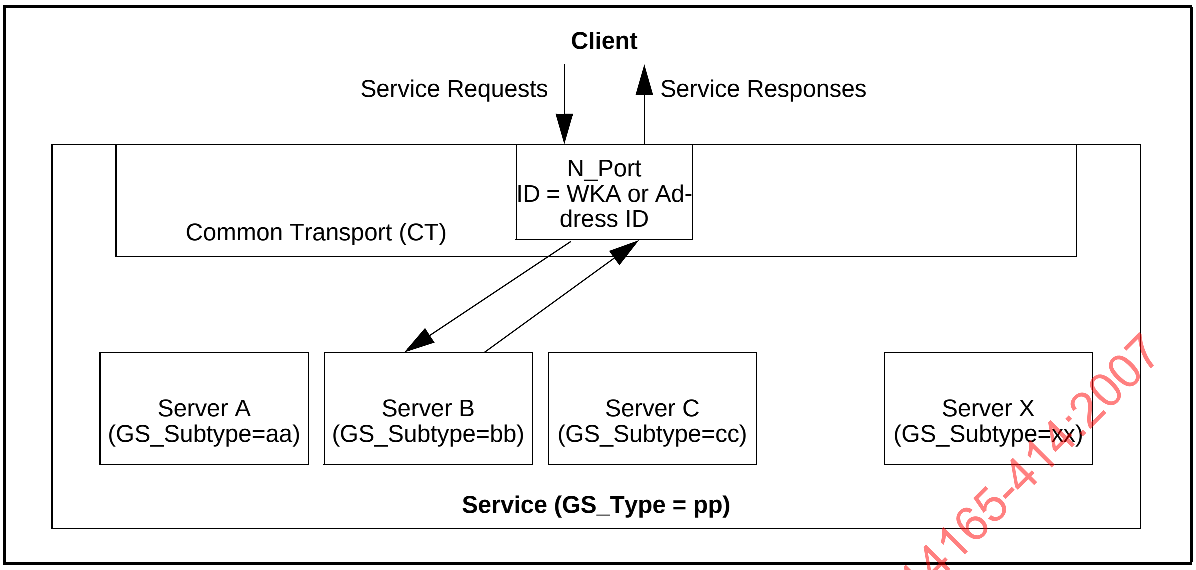


Figure 2 – Relationship between Common Transport, Service, and Servers

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) mode of transaction;

d) Class of service and

e) maximum size of an IU.

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

a) Directory Service;

b) Alias Service;

c) Management Service and

d) Time Service.

There are three types of transactions:

a) Request: where one entity (Client) transmits an IU to another entity (Server) to request a Service;

b) Response: where the Server transmits an IU to the Client responding to its earlier request for a Service and

c) Unsolicited: where one entity (either a Client or a Server) transmits an IU to another entity about an event.

There are two modes of transaction:

a) Asynchronous: in which a Client may transmit multiple requests without having to wait for the responses; an unsolicited IU is transmitted under this mode since there is no required response to an unsolicited IU (see 4.5.3) and

b) Synchronous: in which an Client shall not transmit another request until the corresponding response has been received or there is an indication of non-response.

NOTE No Services in this Internationsl Standard make use of Asynchronous transactions. Asynchronous transactions may be utilized by Services in future revisions of this standard.

The Class of service is an indication of the quality of service that a Client expects from the underlying transport. The following three Classes of service may be used to request a Service:

a) Class 1,

b) Class 2 or

c) Class 3.

Since not all Classes are necessarily available, this parameter describes a list of Classes of services in a descending order of preference. It is used by the Client to indicate a preference in the order of availability. Class 3 service, if available in the operational environment, shall also be available to the Client.

NOTE 1 The Class of service preference is specified according to the local service interface (see Annex A). Since the Class of service is only meaningful to the local node, this indication is not transported as part of the CT preamble information.

NOTE 2 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-3).

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:

- Extended CT_IU preamble or
- 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

This 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.

NOTE The version was changed to 03h to allow implementations to indicate support of new FC-GS-4 features (e.g., Time Stamp format (see Annex E), 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 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 (hex)	**Description**
00 to 1F	Vendor Specific
20	Reserved for use by FC-SW-3
F7	Reserved for Key Distribution Service
F8	Alias Service
FA	Management Service
FB	Reserved for Time Service
FC	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 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, as shown in table 5.

Table 5 – Options field bits

Bit Position	Description
7	Exchange mapping. A value of zero indicates a single bidirectional Exchange (synchronous mode transactions). A value of one indicates multiple Exchanges (asynchronous mode transactions).
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 know that the response is incomplete or that it is unable to report that knowledge).
5 to 3	For 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 (hex)	**Description**
0001 to 03FF	Request CT_IU
0400 to 05FF	Reserved for FC-SW-3
0600 to 7EFF	Request CT_IU
7F00 to 7FFF	Common Request CT_IU
8001	Reject Response CT_IU
8002	Accept Response CT_IU
E000 to FFFF	Reserved for FC-SW-3
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:

1) 0000h: No Maximum Size indicated; the Accept CT_IU may be any size,

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

3) 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:

1) 0000h: All available information was returned in the Accept CT_IU,

2) 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

3) 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 sent by the CT shall be as defined in 4.8.2.

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). The format of the time stamp is as defined in E.2.5.4.

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 administered vendor identifier.

NOTE The vendor identification shall be one assigned by INCITS. A list of assigned vendor identifications is on the T10 web site (www.T10.org).

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 behaviour, 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 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 (hex)	**Description**
01	Invalid command code
02	Invalid version level
03	Logical error
04	Invalid CT_IU Size
05	Logical busy
07	Protocol error
09	Unable to perform command request
0B	Command not supported
0D	Server Not Available
0E	Session Could not be Established
others	Reserved
FF	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 Server.

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 SSE) 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, are returned.

Table 11 – Reject CT_IU reason code explanations for all Servers

Value (hex)	**Description**
00	No additional explanation
F0	Authorization Exception
F1	Authentication Exception
F2	Data base full
F3	Data base empty
Others	Reserved

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-FS mapping

4.5.1 Overview

The CT maps CT_IUs into the appropriate Fibre Channel constructs.

4.5.2 Synchronous mode transactions

4.5.2.1 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 Identifier where the Service is offered, in the manner that is specified in FC-FS, 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.

4.5.2.2 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.

4.5.2.3 Exchange and sequence management

For synchronous mode transactions, the Requesting_CT shall set the Exchange mapping bit in the Options field to zero.

A single bidirectional Exchange shall be used. The Request CT_IU shall be the first Sequence of an 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 4.5.2.6).

An Exchange created by a Requesting_CT is to be used only for a specific Server, and shall not be shared with another Server within the same or a different Service (i.e., a single Exchange shall not be used for both Name Server and Alias Server requests).

Each CT_IU shall be mapped into a Sequence. The CT_IU table for synchronous mode transactions are shown in table 12.

Table 12 – CT_IU table for synchronous transaction

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.

4.5.2.4 FC-2 interface

4.5.2.4.1 Overview

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

4.5.2.4.2 Routing bits

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

4.5.2.4.3 Information category

This parameter shall be set as indicated in table 12.

4.5.2.4.4 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 Identifier where the Service is offered. For Response CT_IUs, this is address identifier of the requesting Nx_Port.

4.5.2.4.5 S_ID

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

4.5.2.4.6 Type

The CT shall set this parameter to Fibre Channel services (20h).

4.5.2.4.7 First sequence

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

4.5.2.4.8 Last sequence

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

4.5.2.4.9 Sequence Initiative

This parameter shall be set as described in 4.5.2.3.

4.5.2.4.10 Continue sequence condition

The use of this parameter is as defined in FC-FS.

4.5.2.4.11 Exchange reassembly

The CT shall set this parameter to zero.

4.5.2.4.12 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.

4.5.2.4.13 Optional headers

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

4.5.2.5 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 Services that make use of synchronous mode transactions typically correlate a request to a response through the use of Exchange IDs.

4.5.2.6 Error detection and recovery

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

a) invalid CT_IU or FC-CT protocol error;

  1) invalid/undefined GS_Type;

  2) invalid revision level;

  3) invalid Options or

  4) Sequence payload exceeds the maximum size of CT_IU at a destination FC_CT,

  and

b) FC-FS protocol errors and timeouts;

  1) Sequence errors;

  2) request to response timeout; or

  3) 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 protocol error is detected, the Exchange shall be terminated. Synchronous mode 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.

4.5.3 Asynchronous mode transactions

4.5.3.1 Fabric login and N_Port login

The Nx_Port shall perform Fabric Login and shall perform N_Port Login with the WKA of the Service, in the manner that is specified in FC-FS.

4.5.3.2 Class of Service

For asynchronous mode transactions, the Class of Service is not defined by this standard.

NOTE For simplification when using asynchronous transactions, a replying CT should use the same Class of service in its subsequent communication with an initiating CT.

4.5.3.3 Exchange and Sequence management

For asynchronous mode transactions, the Requesting_CT shall set the Exchange mapping bit in the Options field to one.

Separate Exchanges in each direction shall be used (e.g., each CT shall originate an Exchange and hold the Sequence Initiative (SI)). Transfer of the SI to the other CT shall be considered a protocol error and the other CT shall terminate the Exchange.

NOTE Asynchronous transaction mode is not used by any Service defined in this standard. This mode may be removed in a future version of FC-GS.

An Exchange created by a CT is to be used only for a specific Server and shall not be shared with another Server within the same or a different Service (e.g., a single Exchange shall not be used for both Name Server and Alias Server requests).

Each CT_IU shall be mapped into a Sequence. The CT_IU table for asynchronous mode transactions are shown in table 13.

Table 13 – CT_IU table for asynchronous transactions

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

4.5.3.4 FC-2 interface

4.5.3.4.1 Overview

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

4.5.3.4.2 Routing bits

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

4.5.3.4.3 Information category

This parameter shall be set as indicated in table 13.

4.5.3.4.4 D_ID

The D_ID shall identify the destination Fibre Channel address identifier for the CT_IU.

4.5.3.4.5 S_ID

The S_ID shall identify the source Fibre Channel address identifier for the CT_IU.

4.5.3.4.6 Type

The CT shall set this parameter to “Fibre Channel services” (20h).

4.5.3.4.7 First sequence

The CT shall set this parameter true to originate a new Exchange.

4.5.3.4.8 Last sequence

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

4.5.3.4.9 Sequence initiative

This parameter shall be set as described in 4.5.3.3.

4.5.3.4.10 Continue sequence condition

The use of this parameter is vendor-specific.

4.5.3.4.11 Exchange reassembly

The CT shall set this parameter false.

4.5.3.4.12 Relative offset

Relative offset may be used if the underlying FC-FS supports it. 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.

4.5.3.4.13 Optional headers

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

4.5.3.5 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 Services that make use of asynchronous mode transactions typically include a tag value within the additional information to correlate requests with responses.

4.5.3.6 Error detection and recovery

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

a) invalid CT_IU or FC-CT protocol error;

   1) invalid/undefined GS_Type;

   2) invalid revision level;

   3) invalid Options;

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

b) FC-FS protocol errors:

   1) Sequence errors,

   2) 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 protocol error is detected, the asynchronous mode transaction shall be terminated as follows:

a) if the error is detected by the Exchange originator, it shall send the No Operation Link Service Sequence with the last Sequence bit set to the Exchange responder or

b) if the error is detected by the Exchange responder, there are two methods for the responder to terminate the Exchange:

   1) if the Exchange responder has the Sequence Initiative, it shall send the No Operation Link Service as the last Sequence of the Exchange; or

   2) if the Exchange responder does not have the SI, it shall transmit the Abort Exchange Link Service in another Exchange to the destination Nx_Port.

All error conditions shall also be indicated to the Client.

NOTE Asynchronous transaction error recovery may be enhanced in future revisions of this standard.

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 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 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 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. Use of CT Authentication by specific Services and commands within those Services is not defined by this standard.

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

If CT Authentication is invoked in the Request CT_IU of a synchronous mode transaction, the corresponding Reject CT_IU or Accept CT_IU shall also use CT Authentication.

All CT_IUs of an asynchronous mode Exchange shall use CT Authentication if the first CT_IU of the Exchange invokes CT Authentication. All CT_IUs shall not use CT Authentication if the first CT_IU of the Exchange does not invoke 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:

- Authentication failed, or
- Failure that generates a Reject CT_IU occurred prior to authentication.

The CT Authentication mechanism computes, using a specific algorithm and secret key, a hash that represents a message signature of 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 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.

Management and distribution of a common time reference may be accomplished through use of the Time Server (see E.2) or by other means not defined by this standard. Management and distribution of secret keys are accomplished by means not defined by this standard.

The hash is communicated in the Authentication Hash Block field of the extended CT_IU preamble. The value of the hash is calculated using the HMAC algorithm defined in RFC 2104 (see RFC 2104).

HMAC provides a procedure to calculate a hash using any iterated cryptographic hash function. CT Authentication may be effected using the MD5 hash function (see RFC 1321) or the SHA-1 hash function (see SHA-1).

The HMAC algorithm shall be applied to 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. The entire output of the HMAC algorithm shall be placed in the Authentication Hash Block field; HMAC truncated output shall not be used.

4.8.2 Secure association attributes

Each Client shall maintain the attributes specified in table 14 for each secure association it has established.

Table 14 – Secure Association Attributes

Item	**Size** bytes
client_said	4
peer_said	4
peer_id	4
peer_name	8
algorithm_id	4
credentials	64

client_said: The SAID of the Client.

peer_said: The SAID of the peer Client with which the credentials were exchanged.

peer_id: The address identifier of the peer Client with which the credentials were exchanged.

peer_name: The N_Port_Name of the peer Client with which the credentials were exchanged.

algorithm_id: The identifier of the security algorithm for the credentials. See 4.8.3.

credentials: The value of the credentials (e.g., keys, secrets, passwords) used by the security algorithm. The format of this field is outside the scope of this standard.

This information is important to subsequent secured communication. Each party to a Secured Association shall use the other party's SAID to identify the credentials and algorithm used.

4.8.3 Security algorithms

This standard allows the use of security algorithms other than those identified here.

The security algorithm and any associated credentials (e.g., keys, secrets, passwords) shall be a characteristic of the algorithm identifier. Table 15 defines algorithm identifiers for several popular security algorithms including recommended credential lengths. Vendor-specific algorithm identifiers may utilize other security algorithms and credentials.

Table 15 – Algorithm identifiers

encryption_id (hex)	**Algorithm Type**	**Credential Size**
00000001 to 7FFFFFFF	vendor-specific algorithm	vendor-specific up to 512 bits
80000001	HMAC-MD5 (see RFC 1321, RFC 2104)	128 bit key
80000002	HMAC-SHA1 (see SHA-1, RFC 2104)	160 bit key
80030038	Obsolete	---
80040038	Obsolete	---
All others	Reserved	---

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 this request and their responses; fields not specified here shall be set as defined in 4.3.2:

- GS_Type and GS_Subtype shall be set as defined by the server to which the common request is applied; and
- Command Code: see table 16 for Request command codes.

Table 16 – Common Request Command Codes

Code (hex)	**Mnemonic**	**Description**	**Object(s) in Request CT_IU**	**Object(s) in Accept CT_IU**
7FF8	GMI	Get More Information	original Transaction ID Information Offset Fragment ID	Fragment ID Fragment
7FF9	SSB	Server Session Begin	none	none
7FFA	SSE	Server Session End	none	none

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.

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.

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).

The format of the GMI Request CT_IU is shown in table 17. 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 17 – 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.

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 to delineate a session for a client, 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.

Two methods addressed to the server 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 with 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 given by terminating a session by an SSE command.

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 18.

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 18 – 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 FFFF-FAh, management server). The format of the SSE CT_IU Request is shown in table 19.

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 19 – 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 behaviour with respect to successful receipt of both Server Session Begin and End or the absence of either.

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 behaviour of a server with respect to commands received within a server session is specified for each server.

5 Directory service

5.1 Overview

The Directory Service provides a means to discover information about Nodes and 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 20 defines the GS_Subtype codes for the Directory Service.

Table 20 – Directory Service subtype values

Values (hex)	**Description**
01	Obsolete[a]
02	Name Server
03	Obsolete[b]
80 to EF	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 (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. Registrations may be performed by a third party. However, the Name Server may refuse such third party registration for unspecified reasons. Once registered, the attributes are made available to requestors.

Requests for the Name Server are carried over the Common Transport (see clause 4). Three types of requests are defined for the Name Server, as shown in table 21.

Table 21 – Name Server – Request types

Command Code (hex)	**Description**
01xx	Get Object(s) (Query)
02xx	Register Object
03xx	Deregister Object(s)

Table 22 lists the different Fibre Channel objects defined for the Name Server.

Table 22 – 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
NN	Node Name	8-byte Name_Identifier
CS	Class of Service	4-byte bit field, one bit per Class supported
IP	IP Address (Node)	32-bit or 128-bit Internet Protocol address
IPA	Initial Process Associator	8-byte Process_Associator
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)	32-bit or 128-bit Internet Protocol address
FPN	Fabric Port Name	8-byte Name_Identifier
HA	Hard Address	3-byte address identifier
FD	FC-4 Descriptor	variable length (0 to 255-byte) field
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 and queries are managed through protocols containing a set of Request CT_IUs and Response CT_IUs supported by the Name Server.

Synchronous transactions shall be used, as defined in 4.5.2. 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 6.3).

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 response. Fields not specified here shall be set as defined in 4.3.2:

- GS_Subtype: as indicated in table 20, and
- Command Code: see table 23 for Request command codes.

Table 23 – Name Server – Request Command Codes

Code (hex)	**Mnemonic**	**Description**	**Object(s) in Request CT_IU**	**Object(s) in Accept CT_IU**
0100	GA_NXT	Get all next	Port Identifier	Various
0101	GID_A	Get identifiers - scope	none[b]	none[b]
0112	GPN_ID	Get Port Name	Port Identifier	Port Name
0113	GNN_ID	Get Node Name - Port Identifier	Port Identifier	Node Name
0114	GCS_ID	Get Class of Service	Port Identifier	Class of Service
0117	GFT_ID	Get FC-4 TYPEs	Port Identifier	FC-4 TYPEs
0118	GSPN_ID	Get Symbolic Port Name	Port Identifier	Symbolic Port Name
011A	GPT_ID	Get Port Type	Port Identifier	Port Type
011B	GIPP_ID	Get IP Address (Port) - Port Identifier	Port Identifier	IP Address (Port)

Table 23 – Name Server – Request Command Codes (Continued)

Code (hex)	Mnemonic	Description	Object(s) in Request CT_IU	Object(s) in Accept CT_IU
011C	GFPN_ID	Get Fabric Port Name - Port Identifier	Port Identifier	Fabric Port Name
011D	GHA_ID	Get Hard Address - Port Identifier	Port Identifier	Hard Address
011E	GFD_ID	Get FC-4 Descriptors - Port Identifier	Port Identifier	List of FC-4 Descriptors
011F	GFF_ID	Get FC-4 Features - Port Identifier	Port Identifier	FC-4 Features
0121	GID_PN	Get Port Identifier - Port Name	Port Name	Port Identifier
012B	GIPP_PN	Get IP Address (Port) - Port Name	Port Name	IP Address (Port)
0131	GID_NN	Get Port Identifiers - Node Name	Node Name	List of Port Identifiers
0132	GPN_NN	Get Port Names - Node Name	Node Name	List of Port Identifiers and Port Names
0135	GIP_NN	Get IP Address (Node)	Node Name	IP Address (Node)
0136	GIPA_NN	Get Initial Process Associator - Node Name	Node Name	Initial Process Associator
0139	GSNN_NN	Get Symbolic Node Name	Node Name	Symbolic Node Name
0153	GNN_IP	Get Node Name - IP Address (Node)	IP Address (Node)	Node Name
0156	GIPA_IP	Get Initial Process Associator - IP Address (Node)	IP Address (Node)	Initial Process Associator
0171	GID_FT	Get Port Identifiers - FC-4 TYPE	none[a]	List of Port Identifiers
0172	GPN_FT	Get Port Names - FC-4 TYPE	none[a]	List of Port Identifiers and Port Names
0173	GNN_FT	Get Node Names - FC-4 TYPE	none[a]	List of Port Identifiers and Node Names
01A1	GID_PT	Get Port Identifiers - Port Type	Port Type	List of Port Identifiers
01B1	GID_IPP	Get Port Identifiers - IP Address (Port)	IP Address (Port)	Port Identifier

Table 23 – Name Server – Request Command Codes (Continued)

Code (hex)	**Mnemonic**	**Description**	**Object(s) in Request CT_IU**	**Object(s) in Accept CT_IU**
01B2	GPN_IPP	Get Port Name - IP Address (Port)	IP Address (Port)	Port Name
01C1	GID_FPN	Get Port Identifiers - F_Port_Name	Fabric Port Name	List of Port Identifiers
1D1	GPPN_ID	Get Permanent Port Name - Port Identifier	Port Identifier	Permanent Port Name
01F1	GID_FF	Get Port Identifiers - FC-4 Features	FC-4 Features	List of Port Identifiers
01F2	GID_DP	Get Port Identifier - Domain/Port	Domain/Port	Port Identifier
0212	RPN_ID	Register Port Name	Port Identifier, Port Name	none
0213	RNN_ID	Register Node Name	Port Identifier, Node Name	none
0214	RCS_ID	Register Class of Service	Port Identifier, Class of Service	none
0217	RFT_ID	Register FC–4 TYPEs	Port Identifier, FC-4 TYPEs	none
0218	RSPN_ID	Register Symbolic Port Name	Port Identifier, Symbolic Port Name	none
021A	RPT_ID	Register Port Type	Port Identifier, Port Type	none
021B	RIPP_ID	Register IP Address (Port) - Port Identifier	Port Identifier, IP Address (Port)	none
021D	RHA_ID	Register Hard Address - Port Identifier	Port Identifier, Hard Address	none
021E	RFD_ID	Register FC-4 Descriptors - Port Identifier	Port Identifier, FC-4 TYPEs, FC-4 Descriptors	none
021F	RFF_ID	Register FC-4 Features - Port Identifier	Port Identifier, FC-4 Features	none
0235	RIP_NN	Register IP Address (Node)	Node Name, IP Address (Node)	none

Table 23 – Name Server – Request Command Codes (Continued)

Code (hex)	Mnemonic	Description	Object(s) in Request CT_IU	Object(s) in Accept CT_IU
0236	RIPA_NN	Register Initial Process Associator	Node Name, Initial Process Associator	none
0239	RSNN_NN	Register Symbolic Node Name	Node Name, Symbolic Node Name	none
0300	DA_ID	De-register all	Port Identifier	none
7FF9	SSB	Server Session Begin[c]	see 4.9	see 4.9
7FFA	SSE	Server Session End[c]	see 4.9	see 4.9

[a] The FC-4 TYPE is specified as an encoded value, not as an object.

[b] The GID_A request specifies a scope in the request, and the response contains a list of Domain_IDs or Domain_ID/Area_IDs.

[c] The SSB and SSE Requests are two of a set of common CT_IU commands. The details of these Requests are presented in 4.9. Interaction of these requests with the name server is 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 23.

The Name Server may reject registrations:

- due to Name Server resource limitations;
- of Name Server objects associated with Alias addresses or
- 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 registrations of Name Server objects associated with known Alias addresses such as:

- Hunt group identifiers (see FC-FS) or
- Multicast group identifiers (see FC-FS).

However, the Name Server shall not be required to know all Alias addresses nor be required to validate registration requests with the Alias Server.

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

- the address identifier 00 00 00h,
- WKA identifiers or
- 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:

- Port Type;
- Port Identifier;
- Port Name;
- Node Name;
- Class of Service or
- Permanent Port Name.

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

If overlapping registrations for the same object are performed, then the Name Server shall, when all registrations have completed, leave the object as one of the registered object values. However, it is indeterminate which of the overlapping registration requests take precedence.

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 23.

5.2.2.5 Deregistratione

Only one deregistration request is defined for the Name Server. The requests defined for the Name Server are all summarized in table 23.

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

5.2.2.6 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 that 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 22 are described in 5.2.3. A null value is defined for each Name Server object. This value is used when the Name Server needs to return an Accept CT_IU but no value has been registered for the requested Name Server object.

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, as used by the Name Server, shall be identical to the address identifier format defined in FC–PH.

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, as used by the Name Server, shall be identical to the Name_Identifier format defined in FC–PH.

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, as used by the Name Server, shall be identical to the Name_Identifier format defined in FC–PH.

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 – Class 1
0 = Class 1 is not supported by the Port Identifier.
1 = Class 1 is supported by the Port Identifier.

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– Class 4

0 = Class 4 is not supported by the Port Identifier.
1 = Class 4 is supported by the Port Identifier.

Bit 5: reserved

Bit 6 – Class 6

0 = Class 6 is not supported by the Port Identifier.
1 = Class 6 is supported by the Port Identifier.

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 IP Address – Format

Both 32 bit (IPv4) and 128 bit (IPv6) Internet Protocol (IP) address formats may be supported by the Name Server. This object description shall apply to both the IP Address (Port) and the IP Address (Node) objects. However, a Registration for IP Address (Port) shall not affect a value registered for IP Address (Node) and a Registration for IP Address (Node) shall not affect a value registered for IP Address (Port).

The format of the 32 bit (IPv4) IP address, as used by the Name Server, shall use big endian bit and byte order, within a word, and shall be preceded by a 00 00 00 00 00 00 00 00 00 00 FF FFh prefix. This is the format specified in RFC 2373. For example, the Name Server format of the 32 bit (IPv4) IP address 198.53.144.31 is 00 00 00 00 00 00 00 00 00 00 FF FF C6 35 90 1Fh. This is broken into words as shown below:

a) Word 0 shall contain the most significant word of the 128 bit IP address (00 00 00 00h);

b) Word 1 shall contain the second most significant word of the 128 bit IP address (00 00 00 00h);

c) Word 2 shall contain the second least significant word of the 128 bit IP address (00 00 FF FFh); and

d) Word 3 shall contain the least significant word of the 128 bit IP address (C6 35 90 1Fh).

The format of the 128 bit (IPv6) IP address, as used by the Name Server, shall use big endian bit and byte order, within a word. This is the format specified in RFC 2373. For example, the Name Server format of the 128 bit (IPv6) IP address 1080:0:0:0:8:800:200C:417A is 10 80 00 00 00 00 00 00 00 08 08 00 20 0C 41 7Ah. This is broken into words as shown below:

a) Word 0 shall contain the most significant word of the 128 bit IP address (10 80 00 00h);

b) Word 1 shall contain the second most significant word of the 128 bit IP address (00 00 00 00h);

c) Word 2 shall contain the second least significant word of the 128 bit IP address (00 08 08 00h) and

d) Word 3 shall contain the least significant word of the 128 bit IP address (20 0C 41 7Ah).

The null value for the IP Address (Port) and IP Address (Node) Name Server object types is 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00h.

5.2.3.7 Initial Process Associator – Format

The format of the Initial Process Associator, as used by the Name Server, shall be identical to the Process_Associator format defined in FC–PH.

The null value for the Initial Process Associator object is FF FF FF FF FF FF FF FFh.

5.2.3.8 FC–4 TYPEs – Format

The FC-4 TYPEs object indicates a set of supported data structure type values (i.e., supported values of the TYPE field in frame headers) for Device_Data and FC-4 Link_Data frames (see FC-FS). For each possible TYPE value, this standard specifies a flag bit in the FC-4 TYPEs object that indicates whether or not frames with that TYPE value are 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 24 – 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 24):

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

1) Word 0 contains information related to TYPE value 00h through 1Fh,

2) Word 1 contains information related to TYPE value 20h through 3Fh,

3) Word 2 contains information related to TYPE value 40h through 5Fh,

4) Word 3 contains information related to TYPE value 60h through 7Fh,

5) Word 4 contains information related to TYPE value 80h through 9Fh,

6) Word 5 contains information related to TYPE value A0h through BFh,

7) Word 6 contains information related to TYPE value C0h through DFh and

8) Word 7 contains information related to TYPE value E0h through FF;.

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 Fibre Channel Services (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.

5.2.3.9 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.

5.2.3.10 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.

5.2.3.11 Port Type – Format

The format of the Port Type object is a one byte value encoded as shown in table 25.

Table 25 – Port TYPE – encoding

Encoded value (hex)	**Description**
00	Unidentified
01	N_Port
02	NL_Port
03	F/NL_Port
7F	Nx_Port
81	F_Port
82	FL_Port
84	E_Port
85	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.

5.2.3.12 Fabric Port Name – Format

The format of the Fabric Port Name object, as used by the Name Server, shall be identical to the Name_Identifier format defined in FC-FS. The Fabric Port Name for a given Port Identifier is the Port_Name for the Fx_Port to which the Port is attached.

NOTE 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.

5.2.3.13 Hard Address – Format

The format of the Hard Address object, as used by the Name Server, shall be identical to the format of Hard Address defined in the Discover Address (ADISC) Extended Link Service (see FC-FS).

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

5.2.3.14 FC-4 Descriptor – Format

The FC-4 Descriptor object is of variable length, with a minimum of 0 and a maximum of 255 bytes. The contents of these bytes are not defined by this standard and shall not be restricted by the Name Server. More than one FC-4 Descriptor may be associated with a Port Identifier - the FC-4 Descriptors in an FC-4 Descriptor object are selected by the FC-4 TYPEs object. Specific FC-4s may define the contents of this object.

If a FC-4 Descriptor is not registered then the FC-4 Descriptor defaults to a 0 byte length object.

5.2.3.15 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 26).

a) The 5 most significant bits of the TYPE field shall be used to identify the word for the FC–4 Features' object, as follows:

   1) Word 0 contains information related to TYPE code 00h through 07h;

   2) Word 1 contains information related to TYPE code 08h through 0Fh;

   3) Word 2 contains information related to TYPE code 10h through 17h

   4) and so forth to Word 31 that contains information related to TYPE code F8h through 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 26).

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 It is intended that the FC-4 corresponding to the TYPE value define the meaning of the 4-bit field. See Annex B 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 26 – 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]

5.2.3.16 Permanent Port Name – Format

The Permanent Port Name is the Name_Identifier associated with a physical Nx_Port. If multiple Name_Identifier are associated with a single Nx_Port via FDISC (see FC-FS) the Permanent Port Name is the original Name_Identifier associated with that Nx_Port at Fabric Login.

The format of the Permanent Port Name object, as used by the Name Server, shall be identical to the Name_Identifier format defined in FC–FS.

NOTE 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.

5.2.4 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 27 is returned.

Table 27 – Reject CT_IU Reason code explanations

Encoded value (hex)	**Description**
00	No additional explanation
01	Port Identifier not registered
02	Port Name not registered
03	Node Name not registered
04	Class of Service not registered
05	IP Address (Node) not registered
06	Initial Process Associator not registered
07	FC–4 TYPEs not registered
08	Symbolic Port Name not registered
09	Symbolic Node Name not registered
0A	Port Type not registered
0B	IP Address (Port) not registered

Table 27 – Reject CT_IU Reason code explanations

Encoded value (hex)	**Description**
0C	Fabric Port Name not registered
0D	Hard Address not registered
0E	FC-4 Descriptor not registered
0F	FC-4 Features not registered
10	Access denied
11	Unacceptable Port Identifier
12	Data base empty
13	No object registered in the specified scope
14	Domain ID not present
15	Port number not present
16	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 that 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 23.

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 No information is returned for WKAs, Domain Controller addresses, or Alias 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 28. 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 28 – 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 29. 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 Initial Process Associator field returns the registered value for the Initial Process Associator, or the null value if no Initial Process Associator is registered for the Port Identifier.

The IP Address (Node) field returns the registered value for the IP Address (Node), or the null value if no IP Address (Node) is registered for the Node related to 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 IP Address (Port) field returns the registered value for the IP Address (Port), or the null value if no IP Address (Port) 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 29 – 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 to m
Node Name	8
Length of Symbolic Node Name (n)	1
Symbolic Node Name	n
Reserved	255 to n
Initial Process Associator	8
IP Address (Node)	16
Class of Service	4
FC–4 TYPEs	32
IP Address (Port)	16
Fabric Port Name	8
Reserved	1
Hard Address	3

5.2.5.3 Query – Get identifiers (GID_A)

The Name Server shall, when 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 30. The requestor supplies a Domain_ID Scope that defines the scope for which identifiers are sought.

NOTE 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_IDs and Area_IDs within the Domain specified by the Domain_ID Scope corresponding to registered Port Identifiers.

Table 30 – 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 31 and table 32.

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

- 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
- 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 31 – 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 32 – 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	1
Area_ID #1	1
Reserved	1
...	
Control (1 r r r r r r r)	1
Requested Domain_ID #n	1
Area_ID #n	1
Reserved	1

5.2.5.4 Query – Get port name (GPN_ID)

The Name Server shall, when 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 33. The requestor supplies the Port Identifier for which the Port Name is sought.

Table 33 – 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 34.

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

Table 34 – 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, when 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 35. The requestor supplies the Port Identifier for which the Node Name is sought.

Table 35 – 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 36.

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

Table 36 – 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, when 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 37. The requestor supplies the Port Identifier for which the Class of Service object is sought.

Table 37 – 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 38.

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

Table 38 – 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, when 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 39. The requestor supplies the Port Identifier for which the FC–4 TYPEs object is sought.

Table 39 – 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 40.

The FC–4 TYPEs field (see 5.2.3.8) returns the registered value for the FC–4 TYPEs.

Table 40 – 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, when 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 41. The requestor supplies the Port Identifier for which the Symbolic Port Name is sought.

Table 41 – 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 42.

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 42 – 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 to m

5.2.5.9 Query – Get port type (GPT_ID)

The Name Server shall, when 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 43. The requestor supplies the Port Identifier for which the Port Type is sought.

Table 43 – 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 44.

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

Table 44 – 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 IP address (Port) (GIPP_ID)

The Name Server shall, when it receives a GIPP_ID request, return the registered IP Address (Port) for the specified Port Identifier. The format of the GIPP_ID Request CT_IU is shown in table 45. The requestor supplies the Port Identifier for which the IP Address (Port) is sought.

Table 45 – GIPP_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 GIPP_ID request is shown in table 46.

The IP Address (Port) field returns the registered value for the IP Address (Port).

Table 46 – Accept CT_IU to GIPP_ID Request

Item	**Size** bytes
CT_IU preamble	see 4.3
IP Address (Port)	16

5.2.5.11 Query – Get fabric port name (GFPN_ID)

The Name Server shall, when 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 47. The requestor supplies the Port Identifier for which the Fabric Port Name is sought.

Table 47 – 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 48.

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

Table 48 – Accept CT_IU to GFPN_ID Request

Item	**Size** bytes
CT_IU preamble	see 4.3
Fabric Port Name	8

5.2.5.12 Query – Get hard address (GHA_ID)

The Name Server shall, when 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 49. The requestor supplies the Port Identifier for which the Hard Address is sought.

Table 49 – 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 50.

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

Table 50 – Accept CT_IU to GHA_ID Request

Item	**Size** bytes
CT_IU preamble	see 4.3
Reserved	1
Hard Address	3

5.2.5.13 Query – Get FC-4 descriptors (GFD_ID)

The Name Server shall, when it receives a GFD_ID request, return the registered FC-4 Descriptor(s) for the specified Port Identifier and FC-4 TYPEs. The format of the GFD_ID Request CT_IU is shown in table 51. The requestor supplies the Port Identifier and FC-4 TYPEs for which the FC-4 Descriptor(s) are sought.

Table 51 – GFD_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 Accept CT_IU to a GFD_ID request is shown in table 52. The Accept CT_IU shall contain the number of FC-4 Descriptors requested (as indicated by the FC-4 TYPEs field). The FC-4 Descriptors shall be returned in ascending TYPE order.

The Descriptor length field shall contain a single byte unsigned value indicating the size of the variable length FC-4 Descriptor.

The FC-4 Descriptor field returns the registered FC-4 Descriptor for the specified Port Identifier and FC-4 TYPE.

Table 52 – Accept CT_IU to GFD_ID Request

Item	**Size** bytes
CT_IU preamble	see 4.3
Descriptor length (m) #1	1
FC-4 Descriptor #1	m
Reserved	255 to m
...	
Descriptor length (m) #n	1
FC-4 Descriptor #n	m
Reserved	255 to m

NOTE There is no distinction between a FC-4 Descriptor that has not been registered and an FC-4 Descriptor that has been registered with zero length.

5.2.5.14 Query – Get FC–4 Features (GFF_ID)

The Name Server shall, when 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 53. The requestor supplies the Port Identifier for which the FC–4 Features' object is sought.

Table 53 – 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 54.

The FC–4 Features field (see 5.2.3.15) returns the registered value for the FC–4 Features.

Table 54 – Accept CT_IU to GFF_ID Request

Item	Size bytes
CT_IU preamble	see 4.3
FC–4 Features	128

5.2.5.15 Query – Get port identifier (GID_PN)

The Name Server shall, when 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 55. The requestor supplies the Port Name for which the Port Identifier is sought.

Table 55 – 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 56.

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

Table 56 – Accept CT_IU to GID_PN Request

Item	Size bytes
CT_IU preamble	see 4.3
Reserved	1
Port Identifier	3

5.2.5.16 Query – Get IP address (Port) (GIPP_PN)

The Name Server shall, when it receives a GIPP_PN request, return the registered IP Address (Port) for the specified Port Name. The format of the GIPP_PN Request CT_IU is shown in table 57. The requestor supplies the Port Name for which the IP Address (Port) is sought.

Table 57 – GIPP_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 GIPP_PN request is shown in table 58.

The IP Address (Port) field returns the registered value for the IP Address (Port).

Table 58 – Accept CT_IU to GIPP_PN Request

Item	**Size** bytes
CT_IU preamble	see 4.3
IP Address (Port)	16

5.2.5.17 Query – Get Port Identifiers (GID_NN)

The Name Server shall, when 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 59. The requestor supplies the Node Name for which associated Port Identifiers are sought.

Table 59 – 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 60.

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 60 – 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.

5.2.5.18 Query – Get port names (GPN_NN)

The Name Server shall, when 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 61. The requestor supplies the Node Name for which associated Port Identifiers and Port Names are sought.

Table 61 – 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 62.

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

- 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

- 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 62 – 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

5.2.5.19 Query – Get IP address (Node) (GIP_NN)

The Name Server shall, when it receives a GIP_NN request, return the registered IP Address (Node) for the specified Node Name. The format of the GIP_NN Request CT_IU is shown in table 63. The requestor supplies the Node Name for which the IP Address (Node) is sought.

Table 63 – GIP_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 GIP_NN request is shown in table 64.

The IP Address (Node) field returns the registered value for the IP Address (Node).

Table 64 – Accept CT_IU to GIP_NN Request

Item	Size bytes
CT_IU preamble	see 4.3
IP Address (Node)	16

5.2.5.20 Query – Get initial process associator (GIPA_NN)

The Name Server shall, when it receives a GIPA_NN request, return the registered Initial Process Associator object for the specified Node Name. The format of the GIPA_NN Request CT_IU is shown in table 65. The requestor supplies the Node Name for which the Initial Process Associator is sought.

Table 65 – GIPA_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 GIPA_NN request is shown in table 66.

The Initial Process Associator field returns the registered Initial Process Associator object for the specified Node Name.

Table 66 – Accept CT_IU to GIPA_NN Request

Item	Size bytes
CT_IU preamble	see 4.3
Initial Process Associator	8

5.2.5.21 Query – Get symbolic node name (GSNN_NN)

The Name Server shall, when 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 67. The requestor supplies the Node Name for which the Symbolic Node Name is sought.

Table 67 – 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 68.

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 68 – 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 to *n*

5.2.5.22 Query – Get node name (GNN_IP)

The Name Server shall, when it receives a GNN_IP request, return the registered Node Name object for the specified IP Address (Node). The format of the GNN_IP Request CT_IU is shown in table 69. The requestor supplies the IP Address (Node) for which the Port Name is sought.

Table 69 – GNN_IP Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
IP Address (Node)	16

The format of the Accept CT_IU to a GNN_IP request is shown in table 70.

The Node Name field returns the registered Node Name.

Table 70 – Accept CT_IU to GNN_IP Request

Item	**Size** bytes
CT_IU preamble	see 4.3
Node Name	8

5.2.5.23 Query – Get initial process associator (GIPA_IP)

The Name Server shall, when it receives a GIPA_IP request, return the registered Initial Process Associator object for the specified IP Address (Node). The format of the GIPA_IP Request CT_IU is shown in table 71. The requestor supplies the IP Address (Node) for which the Initial Process Associator is sought.

Table 71 – GIPA_IP Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
IP Address (Node)	16

The format of the Accept CT_IU to a GIPA_IP request is shown in table 72.

The Initial Process Associator field returns the registered value for the Initial Process Associator.

Table 72 – Accept CT_IU to GIPA_IP Request

Item	**Size** bytes
CT_IU preamble	see 4.3
Initial Process Associator	8

5.2.5.24 Query – Get port identifiers (GID_FT)

The Name Server shall, when 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 73. The requestor supplies the FC–4 TYPE code (see FC-FS) for which supporting Port Identifiers are sought.

NOTE 1 The TYPE is specified as an 8-bit encoded FC-FS value, not as an FC-4 TYPE object.

The Domain_ID Scope and Area_ID Scope fields specify the scope of the request. If both the Domain_ID Scope and the Area_ID Scope fields are 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 and the Area_ID Scope field is 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, the Name Server shall return Port Identifiers within the specified Domain and Area having registered support for the specified FC–4 TYPE code.

NOTE 2 Suitable values for the Domain_ID Scope and Area_ID Scope fields may be discovered using the GID_A query.

Table 73 – GID_FT Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
Reserved	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 74.

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

- 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
- 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 74 – 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

5.2.5.25 Query – Get port names (GPN_FT)

The Name Server shall, when 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 75. The requestor supplies the FC–4 TYPE code (see FC-FS) for which supporting Port Identifiers and Port Names are sought.

NOTE 1 The TYPE is specified as an 8-bit encoded FC-FS value, not as an FC-4 TYPE object.

The Domain_ID Scope and Area_ID Scope fields specify the scope of the request. If both the Domain_ID Scope and the Area_ID Scope fields are 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 and the Area_ID Scope field is 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, 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 2 Suitable values for the Domain_ID Scope and Area_ID Scope fields may be discovered using the GID_A query.

Table 75 – GPN_FT Request CT_IU

Item	Size bytes
CT_IU preamble	see 4.3
Reserved	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 76.

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

- 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
- 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 76 – 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

5.2.5.26 Query – Get node names (GNN_FT)

The Name Server shall, when 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 77. The requestor supplies the FC–4 TYPE code (see FC-FS) for which supporting Port Identifiers and Node Names are sought.

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

The Domain_ID Scope and Area_ID Scope fields specify the scope of the request. If both the Domain_ID Scope and the Area_ID Scope fields are 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 and the Area_ID Scope field is 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, 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 Suitable values for the Domain_ID Scope and Area_ID Scope fields may be discovered using the GID_A query.

Table 77 – GNN_FT Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
Reserved	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 78.

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:

- 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

- 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 78 – 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

5.2.5.27 Query – Get port identifiers (GID_PT)

The Name Server shall, when 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 79. The requestor supplies the Port Type for which supporting Port Identifiers are sought.

The Domain_ID Scope and Area_ID Scope fields specify the scope of the request. If both the Domain_ID Scope and the Area_ID Scope fields are 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 and the Area_ID Scope field is 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, the Name Server shall return Port Identifiers within the specified Domain and Area having registered support for the specified Port Type.

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

Table 79 – 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
Reserved	1

The format of the Accept CT_IU to a GID_PT request is shown in table 80.

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

- 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

- 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 80 – 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

5.2.5.28 Query – Get port identifier (GID_IPP)

The Name Server shall, when it receives a GID_IPP request, return the Port Identifier having registered the specified IP Address (Port). The format of the GID_IPP Request CT_IU is shown in table 81. The requestor supplies the IP Address (Port) for which a Port Identifier is sought.

Table 81 – GID_IPP Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
IP Address (Port)	16

The format of the Accept CT_IU to a GID_IPP request is shown in table 82.

Table 82 – Accept CT_IU to GID_IPP Request

Item	**Size** bytes
CT_IU preamble	see 4.3
80h	1
Port Identifier #n	3

NOTE In previous standards this payload was a list of Port Identifiers, but this violates the requirement of uniqueness of IP addresses in a subnet (see RFC 791).

5.2.5.29 Query – Get port name (GPN_IPP)

The Name Server shall, when it receives a GPN_IPP request, return the registered Port Name object for the specified IP Address (Port). The format of the GPN_IPP Request CT_IU is shown in table 83. The requestor supplies the IP Address (Port) for which the Port Name is sought.

Table 83 – GPN_IPP Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
IP Address (Port)	16

The format of the Accept CT_IU to a GPN_IPP request is shown in table 84.

The Port Name field returns the registered Port Name.

Table 84 – Accept CT_IU to GPN_IPP Request

Item	**Size** bytes
CT_IU preamble	see 4.3
Port Name	8

5.2.5.30 Query – Get port identifiers - Fabric port name (GID_FPN)

The Name Server shall, when 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 85. The requestor supplies the Fabric Port Name for which associated Port Identifiers are sought.

Table 85 – 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 86.

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

- 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
- Bits 6 to 0 are reserved.

Table 86 – 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 when there is a single Nx_Port device attached. When 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.

5.2.5.31 Query – Get permanent port name - Port identifier (GPPN_ID)

The Name Server shall, when 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 87. The requestor supplies the Port Identifier for which the Permanent Port Name object is sought.

Table 87 – 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 88.

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

Table 88 – Accept CT_IU to GPPN_ID Request

Item	Size bytes
CT_IU preamble	see 4.3
Permanent Port Name	8

5.2.5.32 Query – Get port identifiers (GID_FF)

The Name Server shall, when 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 89. 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

- Bits 7 to 4 are reserved and
- Bits 3 to 0 contain the four FC-4 Feature bits for the specified TYPE code.

The Domain_ID Scope and Area_ID Scope fields specify the scope of the request. If both the Domain_ID Scope and the Area_ID Scope fields are 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 and the Area_ID Scope field is 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, 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 Suitable values for the Domain_ID Scope and Area_ID Scope fields may be discovered using the GID_A query.

Table 89 – GID_FF Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
Reserved	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 90.

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

- 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
- 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 90 – 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

5.2.5.33 Query – Get port identifier (Domain/Port) (GID_DP)

The Name Server shall, when 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 91. The requestor supplies the Domain_ID and Physical Port Number for which the Port Identifiers are sought.

Table 91 – 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 92. 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

- 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
- 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 92 – 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

5.2.5.34 Register port name (RPN_ID)

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

The Name Server shall accept RPN_ID requests received from the Port with its address identifier equal to the Port Identifier in the Request CT_IU, from the Link Service Facilitator or from the Fabric Controller. Name Server may reject registration of the Port Name from any other source. The Fabric may register the Port Name for a Port Identifier before explicit Fabric Login (FLOGI) has completed.

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

Deregistration may be accomplished by registering a null Port Name (see 5.2.3.3).

The format of the RPN_ID Request CT_IU is shown in table 93.

Table 93 – RPN_ID Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
Reserved	1
Port Identifier	3
Port Name	8

The format of the RPN_ID Accept CT_IU is shown in table 94.

Table 94 – RPN_ID Accept CT_IU

Item	Size bytes
CT_IU preamble	see 4.3

5.2.5.35 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 Name Server shall accept RNN_ID requests received from the Port with its address identifier equal to the Port Identifier in the Request CT_IU, from the Link Service Facilitator or from the Fabric Controller. Name Server may reject registration of the Node Name from any other source. 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 95.

Table 95 – 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 96.

Table 96 – RNN_ID Accept CT_IU

Item	Size bytes
CT_IU preamble	see 4.3

5.2.5.36 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 Name Server shall accept RCS_ID requests received from the Port with its address identifier equal to the Port Identifier in the Request CT_IU, from the Link Service Facilitator or from the Fabric Controller. Name Server may reject registration of the Class of Service from any other source. 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 97.

Table 97 – 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 98.

Table 98 – RCS_ID Accept CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3

5.2.5.37 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 accept RFT_ID requests received from the Port with its address identifier equal to the Port Identifier in the Request CT_IU, from the Link Service Facilitator or from the Fabric Controller. Name Server may reject registration of FC-4 TYPEs from any other source.

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 5.2.3.8).

The format of the RFT_ID Request CT_IU is shown in table 99.

Table 99 – 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 100.

Table 100 – RFT_ID Accept CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3

5.2.5.38 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 may reject registration of the Symbolic Port Name unless the registration is attempted by the Port with its address identifier equal to the Port Identifier in the Request CT_IU.

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 5.2.3.9).

The format of the RSPN_ID Request CT_IU is shown in table 101. The Name length field shall contain a single byte unsigned value indicating the size of the variable length Symbolic Port Name.

Table 101 – 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 102.

Table 102 – RSPN_ID Accept CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3

If the RSPN_ID Name Server request is rejected by the Name Server because the Name length field value does not match the size of the Symbolic Port Name in the Request CT_IU, then the Reject CT_IU reason code shall be ‘Invalid IU Size’, with a reason explanation code of ‘No additional explanation’.

5.2.5.39 Register port type (RPT_ID)

The RPT_ID Name Server request shall be used to associate a Port Type with a given Port Identifier.

The Name Server shall accept RPT_ID requests received from the Port with its address identifier equal to the Port Identifier in the Request CT_IU, from the Link Service Facilitator or from the Fabric Controller. Name Server may reject registration of the Port Type from any other source. The Fabric may register the Port Type for a Port Identifier before explicit Fabric Login (FLOGI) has completed.

The Name Server shall not attempt validation of the Port TYPEs object and shall accept any 8 bit value. Although not precluded by the Name Server, a Port Identifier shall not register its Port Type as an Nx_Port.

Deregistration may be accomplished by registering a null Port Type object (see 5.2.3.11).

The format of the RPT_ID Request CT_IU is shown in table 103.

Table 103 – RPT_ID Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
Reserved	1
Port Identifier	3
Port Type	1
Reserved	3

The format of the RPT_ID Accept CT_IU is shown in table 104.

Table 104 – RPT_ID Accept CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3

5.2.5.40 Register IP address (Port) (RIPP_ID)

The RIPP_ID Name Server request shall be used to associate an IP Address (Port) with a given Port Identifier.

The Name Server shall accept RIPP_ID requests received from the Port with its address identifier equal to the Port Identifier in the Request CT_IU payload, from the Link Service Facilitator or from the Fabric Controller. Name Server may reject registration of the IP Address (Port) from any other source.

The Name Server shall not attempt validation of the IP Address object and shall accept any 128 bit value.

Deregistration may be accomplished by registering a null IP Address (Port) object (see 5.2.3.6).

The format of the RIPP_ID Request CT_IU is shown in table 105.

Table 105 – RIPP_ID Request CT_IU)

Item	**Size** bytes
CT_IU preamble	see 4.3
Reserved	1
Port Identifier	3
IP Address (Port)	16

The format of the RIPP_ID Accept CT_IU is shown in table 106.

Table 106 – RIPP_ID Accept CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3

5.2.5.41 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 Name Server shall accept RHA_ID requests received from the Port with its address identifier equal to the Port Identifier in the Request CT_IU payload, from the Link Service Facilitator or from the Fabric Controller. Name Server may reject registration of the Hard Address from any other source. 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 5.2.3.13).

The format of the RHA_ID Request CT_IU is shown in table 107.

Table 107 – 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 108.

Table 108 – RHA_ID Accept CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3

5.2.5.42 Register FC-4 descriptor (RFD_ID)

The RFD_ID Name Server request shall be used to associate one or more FC-4 Descriptors with a given Port Identifier.

The Name Server may reject registration of the FC-4 Descriptor unless the registration is attempted by the Port with its address identifier equal to the Port Identifier in the Request CT_IU.

The Name Server shall not attempt validation of the FC-4 Descriptor object and shall accept any value up to 255 bytes in length.

Deregistration may be accomplished by registering a null FC-4 Descriptor object (see 5.2.3.14). Registration of an FC-4 Descriptor for a particular FC-4 TYPE shall not be affected by the subsequent registration of an FC-4 Descriptor for a different FC-4 TYPE.

NOTE For example, an FC-4 Descriptor registration for FC-4 TYPE 'A', followed by an FC-4 Descriptor registration for FC-4 TYPE 'B', results in FC-4 Descriptors registered for each FC-4 TYPEs 'A' and 'B'.

The format of the RFD_ID Request CT_IU is shown in table 109. The Request CT_IU shall contain the number of FC-4 Descriptors requested (as indicated by the FC-4 TYPEs field). The FC-4 Descriptors shall be supplied in ascending TYPE order. The Descriptor length field shall contain a single byte unsigned value indicating the size of the variable length FC-4 Descriptor.

Table 109 – RFD_ID Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
Reserved	1
Port Identifier	3
FC–4 TYPEs	32
Descriptor length (m) #1	1
FC-4 Descriptor #1	m
Reserved	255 to m
...	
Descriptor length (m) #n	1
FC-4 Descriptor #n	m
Reserved	255 to m

The format of the RFD_ID Accept CT_IU is shown in table 110.

Table 110 – RFD_ID Accept CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3

If the RFD_ID Name Server request is rejected by the Name Server because the Descriptor length field value does not match the size of the FC-4 Descriptor in the Request CT_IU, then the Reject CT_IU reason code shall be ‘Invalid IU Size’, with a reason explanation code of ‘No additional explanation’.

5.2.5.43 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 is supported by a given Port Identifier.

The Name Server shall accept RFF_ID requests received from the Port with its address identifier equal to the Port Identifier in the Request CT_IU, from the Link Service Facilitator or from the Fabric Controller. Name Server may reject registration of the FC-4 Features from any other source.

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

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 111.

Table 111 – 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 112.

Table 112 – RFF_ID Accept CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3

5.2.5.44 Register IP Address (Node) (RIP_NN)

The RIP_NN Name Server request shall be used to associate an IP Address (Node) with a given Node Name.

Attempts at registration of an IP Address (Node) shall be rejected by the Name Server unless Node Name registration has been successfully completed (see 5.2.5.35). The Name Server may reject registration of the IP Address (Node) unless the registration is attempted by one of the Port Identifiers associated with the Node Name in the Request CT_IU payload.

The Name Server shall not attempt validation of the IP Address object and shall accept any 128 bit value.

Deregistration may be accomplished by registering a null IP Address (Node) object (see 5.2.3.6).

The format of the RIP_NN Request CT_IU is shown in table 113.

Table 113 – RIP_NN Request CT_IU)

Item	Size bytes
CT_IU preamble	see 4.3
Node Name	8
IP Address (Node)	16

The format of the RIP_NN Accept CT_IU is shown in table 114.

Table 114 – RIP_NN Accept CT_IU

Item	Size bytes
CT_IU preamble	see 4.3

If the RIP_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’.

5.2.5.45 Register initial process associator (RIPA_NN)

The RIPA_NN Name Server request shall be used to associate an Initial Process Associator with a given Node Name.

Attempts at registration of an Initial Process Associator shall be rejected by the Name Server unless Node Name registration has been successfully completed (see 5.2.5.35). The Name Server may reject registration of the Initial Process Associator 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 Initial Process Associators object and shall accept any 8 byte value.

Deregistration may be accomplished by registering a null Initial Process Associator object (see 5.2.3.7).

The format of the RIPA_NN Request CT_IU is shown in table 115.

Table 115 – RIPA_NN Request CT_IU

Item	Size bytes
CT_IU preamble	see 4.3
Node Name	8
Initial Process Associator	8

The format of the RIPA_NN Accept CT_IU is shown in table 116.

Table 116 – RIPA_NN Accept CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3

If the RIPA_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’.

5.2.5.46 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 5.2.5.35). 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 5.2.3.10).

The format of the RSNN_NN Request CT_IU is shown in table 117. The Name length field shall contain a single byte unsigned value indicating the size of the variable length Symbolic Node Name.

Table 117 – 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 118.

Table 118 – 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’.

If the RSNN_NN Name Server request is rejected by the Name Server because the Name length field value does not match the size of the Symbolic Node Name in the Request CT_IU, then the Reject CT_IU reason code shall be ‘Invalid IU Size’, with a reason code explanation of ‘No additional explanation’.

5.2.5.47 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 accept DA_ID requests received from the Port with its address identifier equal to the Port Identifier in the Request CT_IU payload, from the Link Service Facilitator or from the Fabric Controller. Name Server may 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 119. The Port Identifier format shall be as defined in 5.2.3.2.

Table 119 – 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 120.

Table 120 – 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’.

6 Management service

The Management Service provides a single management access point within the Fibre Channel fabric. Management Service covers the following areas:

- Fabric Configuration Server - provides for the configuration management of the fabric;
- Unzoned Name Server - provides access to Name Server (see 5.2) information that is not subject to Zone constraints;
- Fabric Zone Server - provides access to, and control of, Zoning;
- Performance Server - provides performance metrics for fabrics;
- Security Policy Server - provides distribution of security policies and
- Fabric Device Management Interface - provides access to data associated with attached devices.

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 Use of CT Authentication (see 4.8) or of other methods to ensure message integrity and authentication (see FC-SP) is recommended for use within the management service.

The GS_Type for all Management Services shall be set as indicated in Table 4. Table 121 defines the GS_Subtype codes for the Management Service.

Table 121 – Management Service subtype values

Values (hex)	**Description**	**Defining sublause**
01	Fabric Configuration Server	see 6.1
02	Unzoned Name Server	see 6.2
03	Fabric Zone Server	see 6.3
04	Reserved for Lock Server	-
05	Reserved for Performance Server	see Annex F
06	Security Policy Server	see 6.4
10-1F	Fabric Device Management Interface	see 6.5
E0-FF	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. This system view by a management application shall not be constrained by the operational environment (Zone) of its associated Node; therefore, some form of authentication and/or access control is desirable.

6.1 Fabric configuration server

6.1.1 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.

6.1.2 Protocol

6.1.2.1 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.

Synchronous transactions shall be used, as defined in 4.5.2. 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.

6.1.2.2 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:

- GS_Subtype: as indicated in table 121; and

- Command Code: see table 122 for Request command codes.

Table 122 – Fabric Configuration Server – Request Command Codes

Code (hex)	**Mnem.**	**Description**	**Attribute(s) in Request CT_IU**	**Attribute(s) in Accept CT_IU**
0100	GTIN	Get Topology Information	Interconnect Element Domain_ID	see 6.1.5.2
0101	GIEL	Get Interconnect Element List	none	List of Interconnect Element Names and Types
0111	GIET	Get Interconnect Element Type	Interconnect Element Name	Interconnect Element Type
0112	GDID	Get Domain Identifier	Interconnect Element Name	Domain Identifier
0113	GMID	Get Management Identifier	Interconnect Element Name	Management Identifier
0114	GFN	Get Fabric Name	Interconnect Element Name	Fabric Name
0115	GIELN	Get Interconnect Element Logical Name	Interconnect Element Name	Interconnect Element Logical Name
0116	GMAL	Get Interconnect Element Management Address List	Interconnect Element Name	Interconnect Element Management Address List
0117	GIEIL	Get Interconnect Element Information List	Interconnect Element Name	Interconnect Element Information List
0118	GPL	Get Port List	Interconnect Element Name	List of Port Names, Port Types, Port TX Types, and Port Module Types
0121	GPT	Get Port Type	Port Name	Port Type
0122	GPPN	Get Physical Port Number	Port Name	Physical Port Number
0124	GAPNL	Get Attached Port Name List	Port Name	List of Attached Port Names
0126	GPS	Get Port State	Port Name	Port State
0127	GPSC	Get Port Speed Capabilities	Port Name	Port Speed Capabilities, Port Operating Speed
0128	GATIN	Get Attached Topology Information	Port Name	Attached Topology Information

Table 122 – Fabric Configuration Server – Request Command Codes (Continued)

Code (hex)	**Mnem.**	**Description**	**Attribute(s) in Request CT_IU**	**Attribute(s) in Accept CT_IU**
130	GSES	Get Switch Enforcement Status	Interconnect Element Name	List of Zoning Enforcement Status Objects
0191	GPLNL	Get Platform Node Name List	Platform Name	List of Platform Node Names
0192	GPLT	Get Platform Type	Platform Name	Platform Type
0193	GPLML	Get Platform Management Address List	Platform Name	Platform Management Address List
0197	GPAB	Get Platform Attribute Block	Platform Name	Platform Attribute Block
01A1	GNPL	Get Platform Name - Node Name	Platform Node Name	Platform Name
01A2	GPNL	Get Platform Name List	none	List of Platform Names
01A4	GPFCP	Get Platform FCP Type	Platform Name	FCP-2 features bit mask
01A5	GPLI	Get Platform OS LUN Mappings	Platform Name	OS LUN map block
01B1	GNID	Get Node Identification Data - Node Name	Platform Node Name	none[a]
0215	RIELN	Register Interconnect Element Logical Name	Interconnect Element Name, Interconnect Element Logical Name	none
0280	RPL	Register Platform	Platform Name, Platform Type, Platform Management Address List, Platform Node Name List	none
0291	RPLN	Register Platform Node Name	Platform Name, Platform Node Name	none
0292	RPLT	Register Platform Type	Platform Name, Platform Type	none
0293	RPLM	Register Platform Management Address	Platform Name, Platform Management Address	none

Table 122 – Fabric Configuration Server – Request Command Codes (Continued)

Code (hex)	Mnem.	Description	Attribute(s) in Request CT_IU	Attribute(s) in Accept CT_IU
0298	RPAB	Register Platform Attribute Block	Platform Name, Platform Attribute Block	none
029A	RPFCP	Register Platform FCP Type	Platform Name, FCP-2 Features Bit Mask	none
029B	RPLI	Register Platform OS LUN Mappings	Platform Name, OS LUN Map Block	none
0380	DPL	Deregister Platform	Platform Name	none
0391	DPLN	Deregister Platform Node Name	Platform Node Name	none
0392	DPLM	Deregister Platform Management Address	Platform Name, Platform Management Address	none
0393	DPLML	Deregister Platform Management Address List	Platform Name	none
0394	DPLI	Deregister Platform OS LUN Mappings	Platform Name	none
0395	DPAB	Deregister Platform Attribute Block	Platform Name	none
039F	DPALL	De-Register All Platform Information	Null	Null
Other		Reserved		

[a] The Accept CT_IU for GNID contains the ACC payload defined for the Request Node Identification Data Extended Link Service.

6.1.2.3 Registration

The registration requests defined for the Fabric Configuration Server are summarized in table 122. 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.

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.

6.1.2.4 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 122.

6.1.3 Fabric configuration server objects and attributes

6.1.3.1 Overview

Figure 3 illustrates the physical fabric, consisting of one or more Interconnect Elements, that each have some number of physical Ports. These Ports are then connected either to other Ports on other Interconnect Elements, or to Nx_Ports outside of the physical fabric.

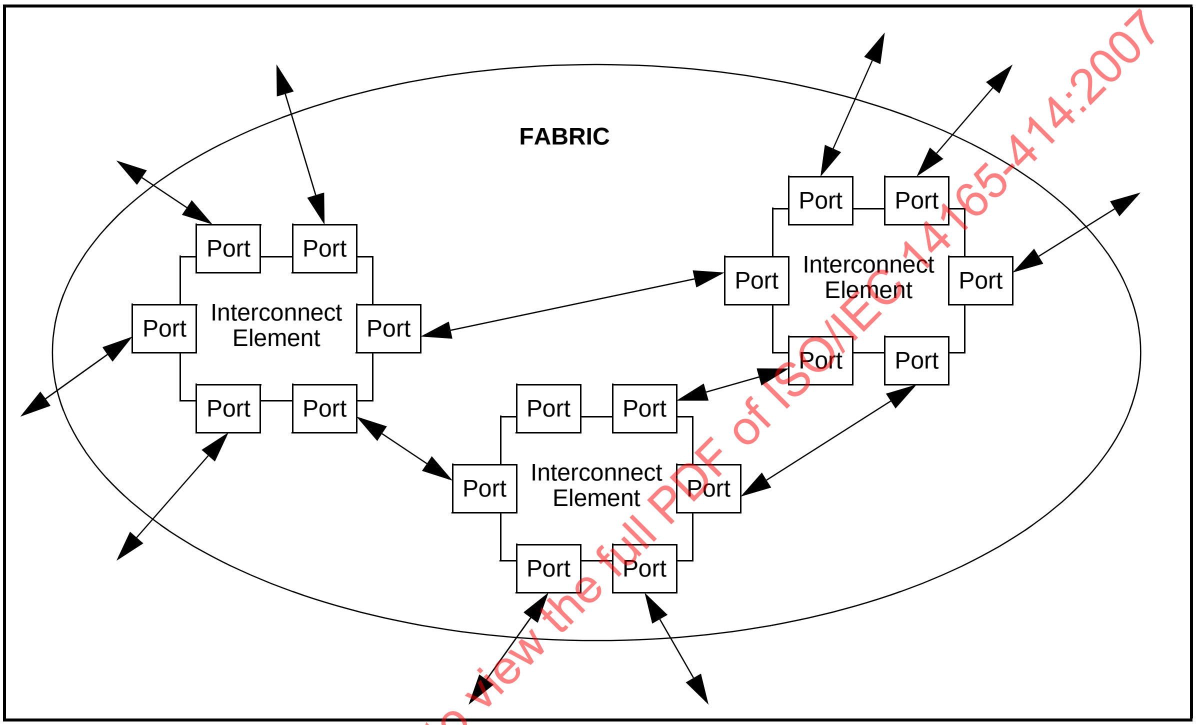


Figure 3 – Physical Fabric Illustration

The Fabric Configuration Server object model is shown in figure 4.

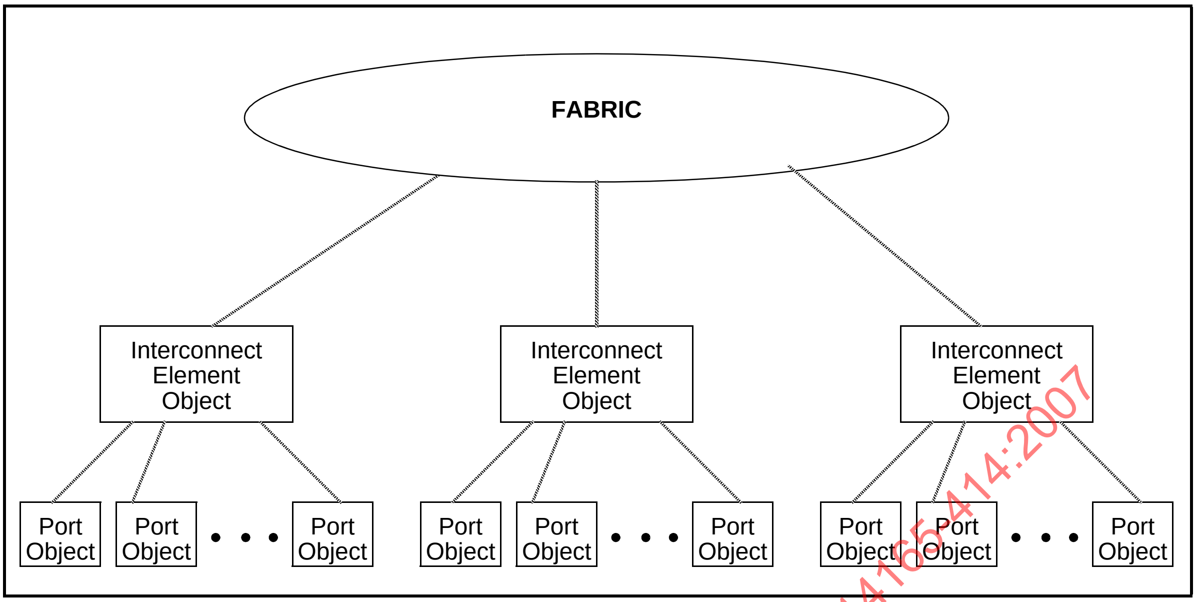


Figure 4 – Fabric Configuration Server Object Model

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 5.

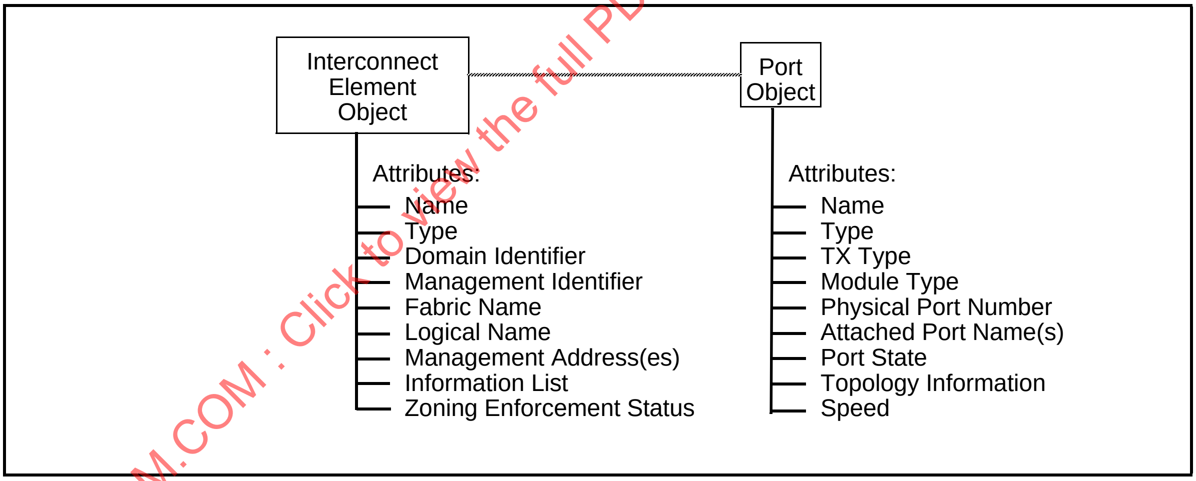


Figure 5 – Interconnect Element and Port attributes

6.1.3.2 Interconnect Element Object

6.1.3.2.1 Interconnect Element Name

The format of the Interconnect Element Name attribute, as used by the Fabric Configuration Server, shall be identical to the Name_Identifier format defined in FC-FS. If the Interconnect Element is a Switch (see FC-SW), 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.

6.1.3.2.2 Interconnect Element Type

The format of the Interconnect Element Type attribute, shall be as shown in table 123.

Table 123 – Interconnect Element Type– encoding

Encoded value (hex)	**Description**
00	Unknown
01	Switch
02	Hub
03	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'.

6.1.3.2.3 Interconnect Element Domain Identifier

The format of the Interconnect Element Domain Identifier attribute, as used by the Fabric Configuration Server, shall be identical to the Domain Identifier format defined in FC-SW-3.

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.

6.1.3.2.4 Interconnect Element Management Identifier

The format of the Interconnect Element Management Identifier attribute, as used by the Fabric Configuration Server, shall be identical to the address identifier format defined in FC–FS. If the Interconnect Element is a Switch (see FC-SW), 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.

6.1.3.2.5 Interconnect Element Fabric Name

The format of the Interconnect Element Fabric Name attribute, as used by the Fabric Configuration Server, shall be identical to the Name_Identifier format defined in FC-FS. The value of the Interconnect Element Fabric Name shall be the same as the value Fabric_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 Interconnect Element Fabric Name attribute is 00 00 00 00 00 00 00 00h.

6.1.3.2.6 Interconnect Element Logical Name

The format of the Interconnect Element Logical Name attribute, as used by the Fabric Configuration Server, shall be as shown in table 124. The contents of these bytes are not defined and shall not be restricted by the Fabric Configuration Server.

Table 124 – Logical Name Format

Item	Size bytes
Logical Name length (m)	1
Logical Name	m
Reserved	255 to m

This attribute may be registered using the protocol described in 6.1.2.3. The null value for the Interconnect Element Logical Name attribute is a zero-length Interconnect Element Logical Name.

6.1.3.2.7 Interconnect Element Management Address

The format of the Interconnect Element Management Address attribute, as used by the Fabric Configuration Server, shall be as shown in table 125. Zero or more Management Address attributes may be associated with an Interconnect Element object.

Table 125 – Management Address Format

Item	Size bytes
Management Address length (m)	1
Management Address value	m
Reserved	255 to 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).

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. The null value for the Interconnect Element Management Address List attribute is a zero-length Interconnect Element Management Address List.

6.1.3.2.8 Interconnect Element Information List

The format of the Interconnect Element Information List attribute, as used by the Fabric Configuration Server, shall be as shown in table 126. This standard does not define how this attribute is registered with the Fabric Configuration Server.

Table 126 – 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.

6.1.3.3 Port Object

6.1.3.3.1 Port Name

The format of the Port Name attribute, as used by the Fabric Configuration Server, shall be identical to the Name_Identifier format defined in FC-FS. 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.

6.1.3.3.2 Port Type

The format of the Port Type attribute shall be as shown in table 127.

Table 127 – Port Type encoding

Encoded value (hex)	**Description**
00	Unidentified
01	N_Port
02	NL_Port
03	F/NL_Port
7F	Nx_Port
81	F_Port
82	FL_Port
84	E_Port
85	B_Port
C0 to FF	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’.

6.1.3.3.3 Port TX Type

The format of the Port TX Type attribute, shall be as shown in table 128.

Table 128 – Port TX Type encoding

Encoded value (hex)	**Description**
01	Unknown
02	Long wave laser - LL (1 550 nm)
03	Short wave laser - SN (850 nm)
04	Long wave laser cost reduced - LC (1 310 nm)
05	Electrical - EL
all others	Reserved

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’.

6.1.3.3.4 Port Module Type

The format of the Port Module Type attribute shall be as shown in table 129.

Table 129 – Port Module Type encoding

Encoded value (hex)	**Description**
01	Unknown
02	Other
03	Obsolete
04	Embedded
05	GLM
06	GBIC with serial ID
07	GBIC without serial ID
08	SFP with Serial ID
09	SFP without Serial ID
0A	XFP
0B	X2 Short
0C	X2 Medium
0D	X2 Tall
0E	XPAK Short
0F	XPAK Medium
10	XPAK Tall
11	XENPAK
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’.

6.1.3.3.5 Physical Port Number

The format of the Physical Port Number attribute, as used by the Fabric Configuration Server, shall be as shown in table 130. 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 130 – 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.

6.1.3.3.6 Attached Port Name

The format of the Attached Port Name attribute, as used by the Fabric Configuration Server, shall be as shown in table 131. Zero or more Attached Port Name attributes may be associated with a Port object.

Table 131 – Attached Port Name Format

Item	**Size** bytes
Port Name	8
Reserved	2
Port Flags	1
Port Type	1

Port Name: As defined in 6.1.3.3.1.

Port Flags: As shown in table 132. .

Table 132 – 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 6.1.3.3.2.

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.

6.1.3.3.7 Port State

The format of the Port State attribute, shall be as shown in table 133.

Table 133 – Port State encoding

Encoded value (hex)	**Description**
00	Unknown
01	Online - a frame may be passed through the Port
02	Offline - a frame is not able to be passed through the Port
03	Testing - port is in a test state
04	Fault - port is not operational
E0 to FF	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’.

6.1.3.3.8 Port Speed Capabilities

The Port Speed Capabilities field identifies a port's data transfer rate capabilities. The format of the Port Speed Capabilities attribute shall be as shown in table 134.

Table 134 – Port Speed Capabilities Format

Item	**Size** bytes
Port Speed Capabilities	2
Reserved	2

All the port'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. Valid bits are as shown in table 135.

Table 135 – Port Speed Capabilities field bits

Bit Position	Description
15	1 Gb/s capable
14	2 Gb/s capable
13	4 Gb/s capable
12	10 Gb/s capable
11 to 1	Reserved
0	Unknown

6.1.3.3.9 Port Operating Speed

The Port Operating Speed field identifies the current operating data transfer rate of a port. The format of the Port Operating Speed attribute shall be as in table 136.

Table 136 – Port Operating Speed Format

Item	**Size** bytes
Reserved	2
Port Operating Speed	2

When a bit is set to one, it indicates the port 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 port's operating speed isn't identifiable, then the "Unknown" bit is set to one. Valid bits are as shown in table 137.

Table 137 – Port Operating Speed field bits

Bit Position	Description
15	1 Gb/s operation
14	2 Gb/s operation
13	4 Gb/s operation
12	10 Gb/s operation
11 to 2	Reserved
1	Unknown
0	Speed not established

6.1.3.3.10 Zoning Enforcement Status Object

The format of the Zoning Enforcement Status Object is depicted in table 138.

Table 138 – 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 switch 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 139.

Table 139 – 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

6.1.3.4 Platform Object

6.1.3.4.1 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 6. (Also see C.4 for an example of the relationship of Platforms to Nodes.)

Platform objects may support multiple FC-4 types, however, the only FC-4 specific platform object currently defined is the FCP-2 Platform object as shown in figure 6.

The Node attributes and the port attributes are identical to the name server objects (see 5.2).

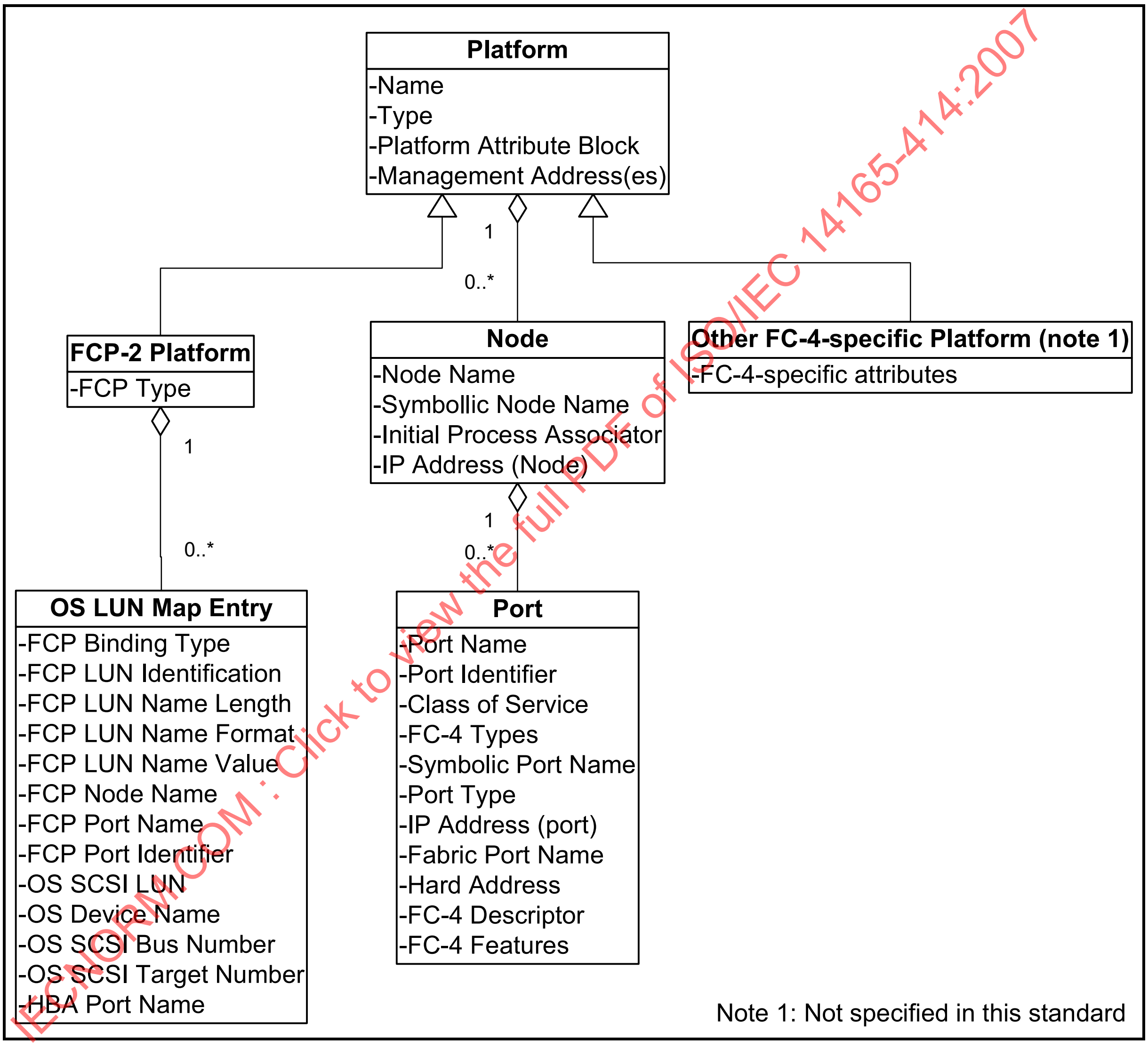


Figure 6 – Platform objects and attributes

6.1.3.4.2 Platform Name

The Platform Name attribute may be registered using the protocol described in 6.1.2.3. The null value for the Platform Name attribute is a zero-length Platform Name.

The format of the Platform Name attribute, as used by the Fabric Configuration Server, shall be as shown in table 140.

Table 140 – Platform Name Format

Item	**Size** bytes
Platform Name length (m)	1
Platform Name	m
Platform Name Format	1
Reserved	254 to 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 141).

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 141).

Table 141 – 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 142.

Table 142 – Code set

Value	**Description**
0h	Reserved
1h	The Platform Name shall contain binary values.
2h	The Platform Name shall contain ASCII graphic codes (i.e., code values 20h through 7Eh)
3h to Fh	Reserved

The Platform Name Type field specifies the format and assignment authority for the platform name, as described in table 143.

Table 143 – 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 are a Vendor ID. The organization associated with the Vendor ID is responsible for ensuring that the remainder of the identifier field is unique.[a]
2h	The Platform Name field contains a canonical form IEEE Extended Unique Identifier, 64-bit (EUI-64). In this case, the identifier length field shall be set to eight. Note that the IEEE guidelines for EUI-64 specify a method for unambiguously encapsulating an IEEE 48-bit identifier within an EUI-64.
3h	The Platform Name field contains an FC-FS Name_Identifier. Any FC-FS identifier may be used, including one of the four based on a Canonical form IEEE company_id.
4h	The Platform Name field contains a UUID. UUIDs are generated and formatted as described in Open Group CAE Specification C309.
5h to Fh	Reserved

[a] The vendor identification shall be one assigned by INCITS. A list of assigned vendor identifications is on the T10 web site (www.T10.org).

6.1.3.4.3 Platform Type

The format of the Platform Type attribute, shall be as shown in table 144.

Table 144 – Platform Type – Encoding

Encoded value (hex)	**Description**
00 00 00 01	Obsolete
00 00 00 02	Other - none of the following
00 00 00 05	Gateway
00 00 00 06	Obsolete
00 00 00 07	Obsolete
00 00 00 08	Obsolete
00 00 00 09	Obsolete
00 00 00 0A	Host
00 00 00 0B	Storage subsystem
00 00 00 0C	Obsolete
00 00 00 0D	Obsolete
00 00 00 0E	Storage access device
00 00 00 0F	Wavelength division multiplexer
00 00 00 11	NAS server
00 00 00 12	Bridge
00 00 00 13	Virtualization device
xx xx xx FF	Multi-function device (see table 145 for values to fill in for xx xx xx)
all others	Reserved

Table 145 – 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 6.1.2.3. The null Platform Type attribute value is set to 00 00 02h ‘Other’.

6.1.3.4.4 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 146.

Table 146 – 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.

6.1.3.4.5 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 147.

Table 147 – Attribute Entry

Item	**Size** bytes
Attribute Entry Type	2
Attribute Entry Length (n)	2
Attribute Entry Value	(see table 148)

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 148.

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 148.

Table 148 – Attribute Entry Types and their associated Values

Type (hex)	Value				
	Description	**Length bytes**	**Type**	**Required**	**Multiples allowed**[a]
0001	Vendor ID	12	ASCII	Yes	No
0002	Product ID	20	ASCII	Yes	No
0003	Product revision level	4 to 32	ASCII	No	No
0004	Description	4 to 128	ASCII	No	No
0005	Label	4 to 64	ASCII	No	No
0006	Location	4 to 128	ASCII	No	No
0007	System ID	4 to 64	ASCII	No	No
0008	System management address	4 to 128	ASCII	No	Yes
0009	Cluster ID	4 to 64	ASCII	No	No
000A	Cluster management address	4 to 128	ASCII	No	Yes
000B	Supported FC-4 Types	20	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 6.1.3.2.7.

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 6.1.3.2.7.

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 5.2.3.8). FCP-2 (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.

6.1.3.4.6 Platform Node Name

The format of the Platform Node Name attribute, as used by the Fabric Configuration Server, shall be identical to the Name_Identifier format defined in FC-FS. 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 6.1.2.3. The null value for the platform Node Name attribute is 00 00 00 00 00 00 00 00h.

6.1.3.4.7 Platform Management Address

The format of the Platform Management Address attribute, as used by the Fabric Configuration Server, is identical to the Interconnect Element Management Address attribute and shall be as defined in 6.1.3.2.7.

6.1.3.4.8 FCP-2 Platform Object

6.1.3.4.8.1 FCP-2 Platform Object Overview

The FCP-2 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.

6.1.3.4.8.2 FCP Type

This is a FC-4 specific attribute for platforms that support FC-4 type of FCP-2. This a 4-byte encoded value, of which two bits are used to indicate the FCP-2 features for the platform (see table 149).

Table 149 – FC-4 Specific Attributes

Bit Position	Description
31 to 4	Reserved
3 to 2	Reserved
1	1 = FCP initiator function supported (see NCITS T10 project FCP-2) 0 = Not supported
0	1 = FCP target function supported (see NCITS T10 project FCP-2) 0 = Not supported

6.1.3.4.9 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 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 150).

Table 150 – 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-2).

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) of the node in the target device that the initiator OS LUN maps to.

FCP Port Name: The Name_Identifier (see FC-FS) of the port in the target device that the initiator OS LUN maps to.

FCP Port Identifier: The port identifier of the 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 port Name_Identifier (see FC-FS) that corresponds to this LUN mapping.

6.1.3.4.10 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.

6.1.3.4.11 Platform Label

An administratively assigned symbolic name for the platform. The Platform Label shall only contain printable ASCII characters.

6.1.3.4.12 Platform Location

The physical location of the platform (e.g., telephone closet, 3rd floor). The Platform Location shall only contain printable ASCII characters.

6.1.4 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 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 151, are returned.

Table 151 – Reject CT_IU Reason code explanations

Encoded value (hex)	**Description**
00	No additional explanation
01	Invalid Name_Identifier for Interconnect Element or Port
10	Interconnect Element List not available
11	Interconnect Element Type not available
12	Domain Identifier not available
13	Management Identifier not available
14	Fabric Name not available
15	Interconnect Element Logical Name not available
16	Management Address List not available
17	Interconnect Element Information List not available
30	Port List not available
31	Port Type not available
32	Physical Port Number not available
34	Attached Port Name List not available
36	Port State not available
50	Unable To register Interconnect Element Logical Name
60	Platform Name does not exist
61	Platform Name already exists
62	Platform Node Name does not exist
63	Platform Node Name already exists.
64 to 6F	Vendor-Specific
70	Platform register operation failed – resource unavailable
71	Zero entries in OS LUN Map
72	Invalid OS Device Name Length
73	Platform Attribute Block Contains Multiple Attributes of the Same Type

Table 151 – Reject CT_IU Reason code explanations (Continued)

Encoded value (hex)	**Description**
74	Invalid Platform Attribute Block Length
75	Required Platform Attributes Not Present
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 that the specified attribute is not available.

6.1.5 Commands

6.1.5.1 Overview

The commands defined for the Fabric Configuration Server are summarized in table 122.

6.1.5.2 Query – Get Topology Information (GTIN)

The Fabric Configuration Server shall, when 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 152.

Table 152 – GTIN Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
Reserved	5
Interconnect Element Domain ID	1
Reserved	6

The format of the Accept CT_IU to a GTIN request is shown in table 153, table 154 and table 155.

Table 153 – 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 154)	68 + (72 x Number of attached Ports)
...	
Topology Information Descriptor #n (*n* = Number of switch Ports that are not E-Ports)	68 + (72 x Number of attached Ports)

Table 154 – 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 127)	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 155 – 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 127)	1
Attached Physical Port Number (see 6.1.2.2.5)	4
Vendor-Specific	28

6.1.5.3 Query – Get Interconnect Element List (GIEL)

The Fabric Configuration Server shall, when 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 156.

Table 156 – GIEL Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3

The format of the Accept CT_IU to a GIEL request is shown in table 157.

Table 157 – 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.

6.1.5.4 Query – Get Interconnect Element Type (GIET)

The Fabric Configuration Server shall, when 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 158.

Table 158 – GIET Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
Interconnect Element Name	8

The format of the Accept CT_IU to a GIET request is shown in table 159.

Table 159 – Accept CT_IU to GIET Request

Item	**Size** bytes
CT_IU preamble	see 4.3
Reserved	3
Interconnect Element Type	1

6.1.5.5 Query – Get Interconnect Element Domain Identifier (GDID)

The Fabric Configuration Server shall, when 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 160.

Table 160 – GDID Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
Interconnect Element Name	8

The format of the Accept CT_IU to a GDID request is shown in table 161.

Table 161 – Accept CT_IU to GDID Request

Item	**Size** bytes
CT_IU preamble	see 4.3
Reserved	1
Interconnect Element Domain Identifier	1
Reserved	2

6.1.5.6 Query – Get Interconnect Element Management Identifier (GMID)

The Fabric Configuration Server shall, when 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 162.

Table 162 – GMID Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
Interconnect Element Name	8

The format of the Accept CT_IU to a GMID request is shown in table 163.

Table 163 – Accept CT_IU to GMID Request

Item	**Size** bytes
CT_IU preamble	see 4.3
Reserved	1
Interconnect Element Management Identifier	3

6.1.5.7 Query – Get Interconnect Element Fabric Name (GFN)

The Fabric Configuration Server shall, when 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 164.

Table 164 – GFN Request CT_IU

Item	Size bytes
CT_IU preamble	see 4.3
Interconnect Element Name	8

The format of the Accept CT_IU to a GFN request is shown in table 165.

Table 165 – Accept CT_IU to GFN Request

Item	Size bytes
CT_IU preamble	see 4.3
Interconnect Element Fabric Name	8

6.1.5.8 Query – Get Interconnect Element Logical Name (GIELN)

The Fabric Configuration Server shall, when 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 166.

Table 166 – GIELN Request CT_IU

Item	Size bytes
CT_IU preamble	see 4.3
Interconnect Element Name	8

The format of the Accept CT_IU to a GIELN request is shown in table 167.

Table 167 – Accept CT_IU to GIELN Request

Item	Size bytes
CT_IU preamble	see 4.3
Interconnect Element Logical Name	256

6.1.5.9 Query – Get Interconnect Element Management Address List (GMAL)

The Fabric Configuration Server shall, when it receives a GMAL request, return all Interconnect Element Management Address attributes for the specified Interconnect Element Name. The format of the GIEL Request CT_IU is shown in table 168.

Table 168 – GMAL Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
Interconnect Element Name	8

The format of the Accept CT_IU to a GMAL request is shown in table 169.

Table 169 – 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.

6.1.5.10 Query – Get Interconnect Element Information List (GIEIL)

The Fabric Configuration Server shall, when 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 170.

Table 170 – GIEIL Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
Interconnect Element Name	8

The format of the Accept CT_IU to a GIEIL request is shown in table 171.

Table 171 – Accept CT_IU to GIEIL Request

Item	**Size** bytes
CT_IU preamble	see 4.3
Interconnect Element Information List	m
Reserved	256 to m

6.1.5.11 Query – Get Port List (GPL)

The Fabric Configuration Server shall, when 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 172.

Table 172 – GPL Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
Interconnect Element Name	8

The format of the Accept CT_IU to a GPL request is shown in table 173.

Table 173 – 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.

6.1.5.12 Query – Get Port Type (GPT)

The Fabric Configuration Server shall, when 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 174.

Table 174 – GPT Request CT_IU

Item	Size bytes
CT_IU preamble	see 4.3
Port Name	8

The format of the Accept CT_IU to a GPT request is shown in table 175.

Table 175 – Accept CT_IU to GPT Request

Item	Size bytes
CT_IU preamble	see 4.3
Reserved	3
Port Type	1

6.1.5.13 Query – Get Physical Port Number (GPPN)

The Fabric Configuration Server shall, when 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 176.

Table 176 – GPPN Request CT_IU

Item	Size bytes
CT_IU preamble	see 4.3
Port Name	8

The format of the Accept CT_IU to a GPPN request is shown in table 177.

Table 177 – Accept CT_IU to GPPN Request

Item	Size bytes
CT_IU preamble	see 4.3
Physical Port Number	4

6.1.5.14 Query – Get Attached Port Name List (GAPNL)

The Fabric Configuration Server shall, when 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 178.

Table 178 – GAPNL Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
Port Name	8

The format of the Accept CT_IU to a GAPNL request is shown in table 179.

Table 179 – 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.

6.1.5.15 Query – Get Port State (GPS)

The Fabric Configuration Server shall, when 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 180.

Table 180 – GPS Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
Port Name	8

The format of the Accept CT_IU to a GPS request is shown in table 181.

Table 181 – 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

6.1.5.16 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 182.

Table 182 – GPSC Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
Port Name	8

The format of the Accept CT_IU to a GPSC is shown in table 183.

Table 183 – GPSC Accept CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
Port Speed Capabilities	2
Port Operating Speed	2

6.1.5.17 Query – Get Attached Topology Information (GATIN)

The Fabric Configuration Server shall, when it receives a GATIN request, return the topology information descriptor (see table 154) for the specified Port Name. The format of the GATIN Request CT_IU is shown in table 184.

Table 184 – GATIN Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
Interconnect Element Port Name	8

The format of the Accept CT_IU to a GATIN request is shown in table 185.

Table 185 – Accept CT_IU to GATIN Request

Item	**Size** bytes
CT_IU preamble	see 4.3
Topology Information Descriptor (see table 154)	68 + (72 x Number of attached Ports)

NOTE 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 (table 6.1.5.11), then issue this command to each Port in turn to discover the additional topologies.

6.1.5.18 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, when 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.

The format of the GSES request payload is depicted in table 186.

Table 186 – 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 187.

Table 187 – 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 138
Zoning enforcement status object #2	see table 138
...	
Zoning enforcement status object #*n*	see table 138

6.1.5.19 Query – Get Platform Node Name List (GPLNL)

The Fabric Configuration Server shall, when 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 188.

Table 188 – GPLNL Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
Platform Name	256

The format of the Accept CT_IU to a GPLNL request is shown in table 189.

Table 189 – 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.

6.1.5.20 Query – Get Platform Type (GPLT)

The Fabric Configuration Server shall, when 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 190.

Table 190 – GPLT Request CT_IU

Item	Size bytes
CT_IU preamble	see 4.3
Platform Name	256

The format of the Accept CT_IU to a GPLT request is shown in table 191.

Table 191 – Accept CT_IU to GPLT Request

Item	Size bytes
CT_IU preamble	see 4.3
Platform Type	4

6.1.5.21 Query – Get Platform Management Address List (GPLML)

The Fabric Configuration Server shall, when 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 192.

Table 192 – GPLML Request CT_IU

Item	Size bytes
CT_IU preamble	see 4.3
Platform Name	256

The format of the Accept CT_IU to a GPLML request is shown in table 193.

Table 193 – 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.

6.1.5.22 Query – Get PLATFORM Attribute Block (GPAB)

The Fabric Configuration Server shall, when it receives a GPAB request, return the Platform Attribute Block for the specified Platform. The GPAB request payload shall specify the Platform Identifier 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.

The format of the GPAB Request payload is depicted in table 194.

Table 194 – 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 6.1.3.4.2.

The format of the GPAB Accept payload is depicted in table 195.

Table 195 – GPAB Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3
Platform Attribute Block	see 6.1.3.4.4

Platform Attribute Block: The format of the Platform Attribute block is described in 6.1.3.4.4.

6.1.5.23 Query – Get Platform Name - Node Name (GNPL)

The Fabric Configuration Server shall, when 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 196.

Table 196 – GNPL Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
Platform Node Name	8

The format of the Accept CT_IU to a GNPL request is shown in table 197.

Table 197 – Accept CT_IU to GNPL Request

Item	**Size** bytes
CT_IU preamble	see 4.3
Platform Name	256

6.1.5.24 Query – Get Platform Name List (GPNL)

The Fabric Configuration Server shall, when 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 198.

Table 198 – GPNL Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3

The format of the Accept CT_IU to a GPNL request is shown in table 199.

Table 199 – 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

6.1.5.25 Query – Get Platform FCP Type (GPFCP)

The Fabric Configuration Server shall, when it receives a GPFCP request, return the FCP-2 Feature Bit Mask (see 6.1.3.4.8.2) for the specified Platform Name. The format of the GPFCP Request CT_IU is shown in table 200.

Table 200 – GPFCP Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
Platform Name	256

The format of the Accept CT_IU to a GPFCP request is shown in table 201.

Table 201 – Accept CT_IU to GPFCP Request

Item	**Size** bytes
CT_IU preamble	see 4.3
FCP-2 Features Bit Mask	4

6.1.5.26 Query – Get Platform OS LUN Mappings (GPLI)

The Fabric Configuration Server shall, when 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 202.

Table 202 – GPLI Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
Platform Name	256

The format of the Accept CT_IU to a GPLI request is shown in table 203.

Table 203 – 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 150.

6.1.5.27 Query – Get Node Identification Data (GNID)

The Fabric Configuration Server shall, when 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 204.

Table 204 – GNID Request CT_IU

Item	Size bytes
CT_IU preamble	see 4.3
Platform Node Name	8
Node Identification Data format	1
Reserved	3

See the Request Node Identification Data ELS in FC-FS for the definition of the Node Identification Data format.

The format of the Accept CT_IU to a GNID request is shown in table 205.

Table 205 – 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 for the definition and length of this payload.

6.1.5.28 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 6.1.3.2.6).

The format of the RIELN Request CT_IU is shown in table 206.

Table 206 – 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 207.

Table 207 – RIELN Accept CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3

6.1.5.29 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 6.1.5.36).

The format of the RPL Request CT_IU is shown in table 208.

Table 208 – 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 209.

Table 209 – RPL Accept CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3

6.1.5.30 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 6.1.5.37).

The format of the RPLN Request CT_IU is shown in table 210.

Table 210 – 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 211.

Table 211 – RPLN Accept CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3

6.1.5.31 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 6.1.3.4.3).

The format of the RPLT Request CT_IU is shown in table 212.

Table 212 – 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 213.

Table 213 – RPLT Accept CT_IU

Item	Size bytes
CT_IU preamble	see 4.3

6.1.5.32 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 6.1.5.39).

The format of the RPLM Request CT_IU is shown in table 214.

Table 214 – 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 215.

Table 215 – RPLM Accept CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3

6.1.5.33 Register Platform Attribute Block (RPAB)

The Fabric Configuration Server shall when 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.

The format of the RPAB Request payload is depicted in table 216.

Table 216 – RPAB Request Payload

Item	**Size** bytes
CT_IU preamble	see 4.3
Platform Name	256
Platform Attribute Block	see 6.1.3.4.4

Platform Name: The format of the Platform Name is described in 6.1.3.4.2.

Platform Attribute Block: The format of the Platform Attribute block is described in 6.1.3.4.4.

The format of the RPAB Accept payload is depicted in table 217.

Table 217 – RPAB Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3

6.1.5.34 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 6.1.3.4.8.2).

The format of the RPFCP Request CT_IU is shown in table 218.

Table 218 – 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 219.

Table 219 – RPFCP Accept CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3

6.1.5.35 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 6.1.5.36).

The format of the RPLI Request CT_IU is shown in table 220.

Table 220 – 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 150.

The format of the RPL Accept CT_IU is shown in table 221.

Table 221 – RPLI Accept CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3

6.1.5.36 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.

The format of the DPL Request CT_IU is shown in table 222.

Table 222 – 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 223.

Table 223 – DPL Accept CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3

6.1.5.37 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.

The format of the DPLN Request CT_IU is shown in table 224.

Table 224 – 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 225.

Table 225 – DPLN Accept CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3

6.1.5.38 Deregister Platform Management Address (DPLM)

The Configuration Server shall, when 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 226). The DPLM accept payload is null (see table 227).

Table 226 – DPLM Request Payload

Item	**Size** bytes
CT_IU Preamble	see 4.3
Platform Name	256
Platform Management Address	256

Table 227 – DPLM Accept Payload

Item	**Size** bytes
CT_IU Preamble	see 4.3

6.1.5.39 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.

The format of the DPLML Request CT_IU is shown in table 228.

Table 228 – 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 229.

Table 229 – DPLML Accept CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3

6.1.5.40 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.

The format of the DPLI Request CT_IU is shown in table 230.

Table 230 – 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 231.

Table 231 – DPLI Accept CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3

6.1.5.41 Deregister Platform Attribute Block (DPAB)

The Fabric Configuration Server shall, when 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.

The format of the DPAB Request payload is depicted in table 232.

Table 232 – 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 233 below.

Table 233 – DPAB Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3

6.1.5.42 De-register all platform information (DPALL)

The Fabric Configuration Server shall, when 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 234.

Table 234 – DPALL Request Payload

Item	**Size** bytes
CT_IU preamble	see 4.3

The format of the DPALL accept payload shall be as shown in table 235.

Table 235 – DPALL Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3

6.2 Unzoned Name Server

6.2.1 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.

6.2.2 Protocol

6.2.2.1 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.

Synchronous transactions shall be used, as defined in 4.5.2. 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 7.

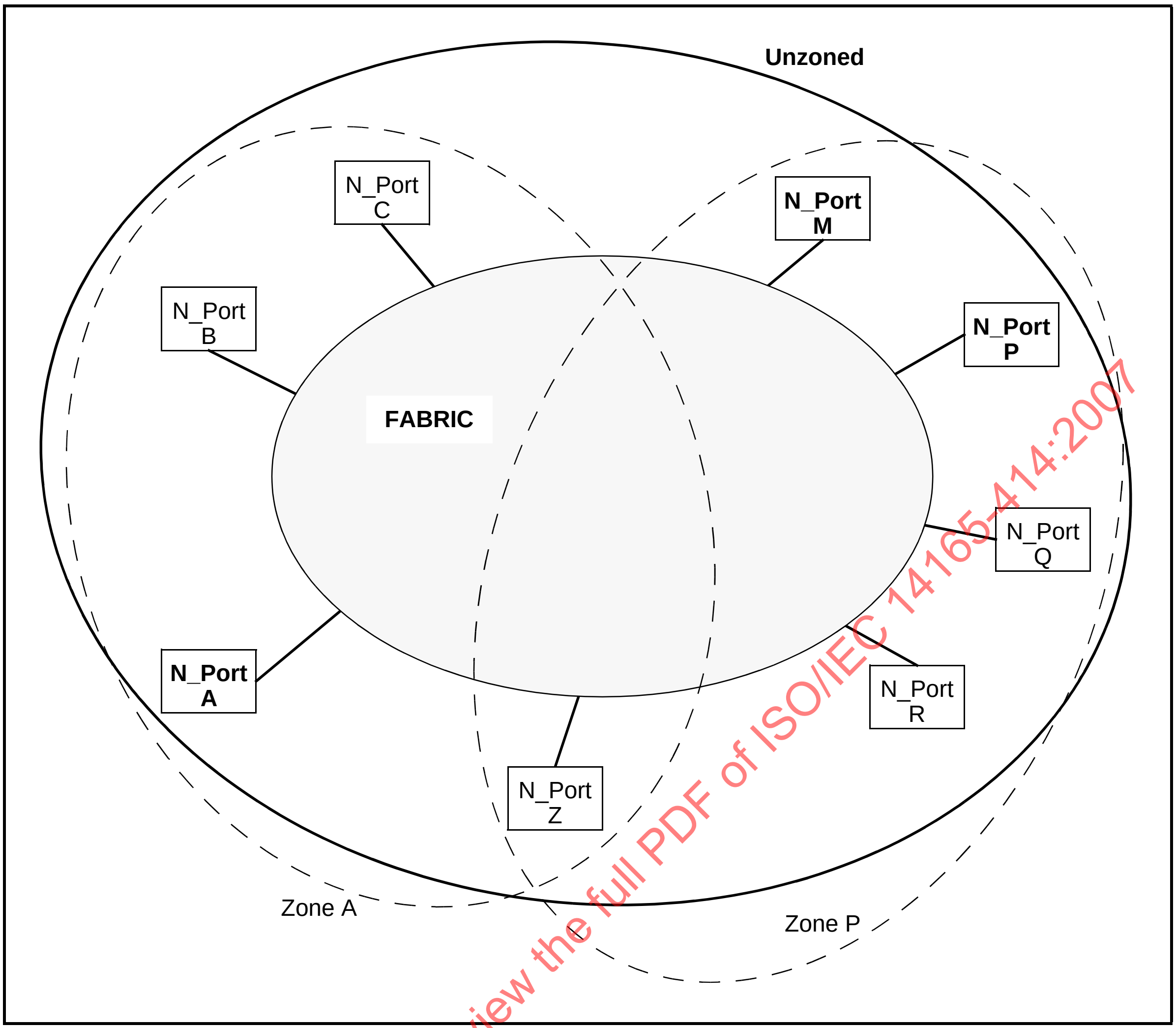


Figure 7 – Name Server Zone Constraints

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:

- Response for N_Port A contains N_Ports B, C, and Z;
- Response for N_Port P contains N_Ports Q, R, M, and Z; and
- 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.

6.2.2.2 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:

- GS_Subtype: as indicated in table 121; and

- Command Code: see table 23 for Request command codes.

6.2.2.3 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’.

6.2.2.4 Queries

Queries shall be performed as defined in 5.2.2.4.

6.2.2.5 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’.

6.2.2.6 Server Sessions

The action of the Unzoned Name Server is unaffected by server sessions.

6.2.3 Unzoned Name Server objects – Formats

Unzoned Name Server objects are as defined in 5.2.3.

6.2.4 Reason code explanations

Unzoned Name Server reason code explanations are as defined in 5.2.4.

6.2.5 Commands

The commands defined for the Unzoned Name Server are summarized in table 23, and are defined in 5.2.5.

6.3 Fabric Zone Server

6.3.1 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

- barriers between devices that use different operating systems. It is often critical to separate servers and storage devices with different operating systems because accidental transfer of information from one to another may delete or corrupt data,
- 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,
- 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
- temporary access between devices for specific purposes. Administrators may remove Zone restrictions temporarily, then restore Zone restrictions to perform normal processes.

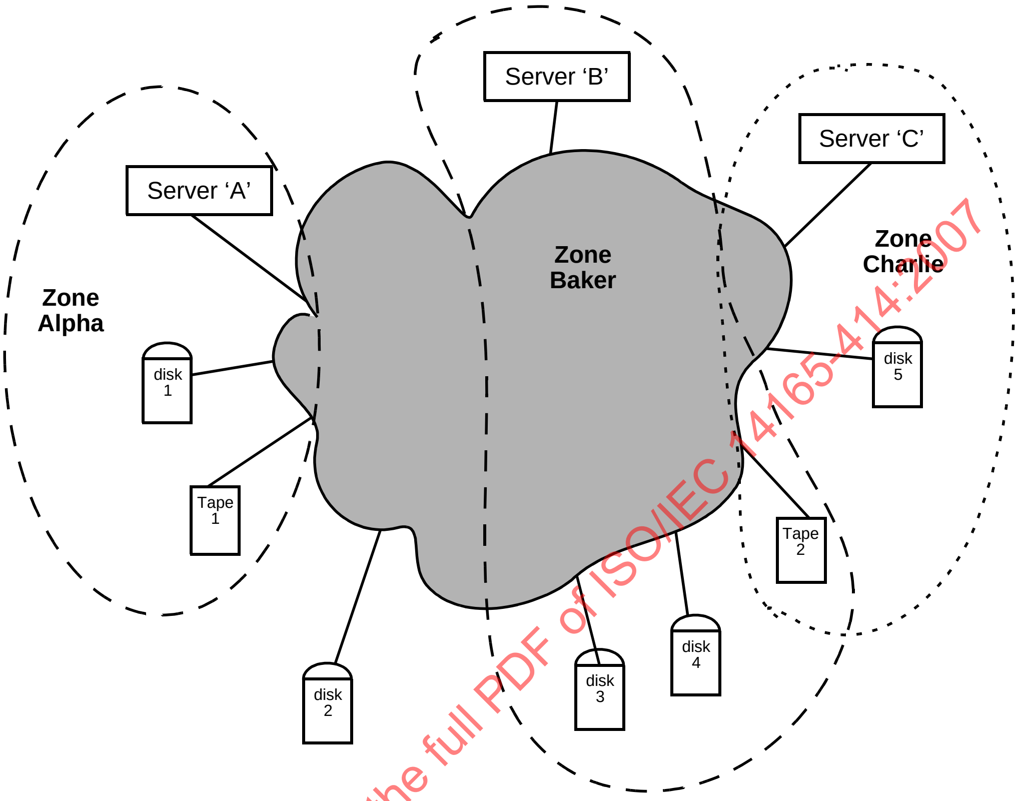
Zones may be configured via this Server or via a mechanism outside the scope of this standard.

- An Nx_Port may be a member of one or more Zones. Zone membership may be specified by
  - the N_Port_Name of the Nx_Port connected to the switch,
  - the N_Port_ID assigned during Fabric Login,
  - the Node_Name associated with the Nx_Port; note that the Node_Name may include more than one Nx_Port,
  - the F_Port_Name of the Fx_Port to which the Nx_Port is connected or
  - the domain identification (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 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 Port Indentifier associated with a Domain_ID and physical port number (see 5.2.5.33).

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 8.



	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 8 – 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-3 to understand how the Zoning information is distributed among the Switches of a multi-switch Fabric.

NOTE An example of Fabric Zone Server usage is given in Annex D.

6.3.2 Terminology

The terminology used in this document is the following:

- Active Zone Set: the Zone Set currently enforced by the Fabric;
- Zone Set Database: the database of the Zone Sets available to be activated within a Fabric and
- 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 9.

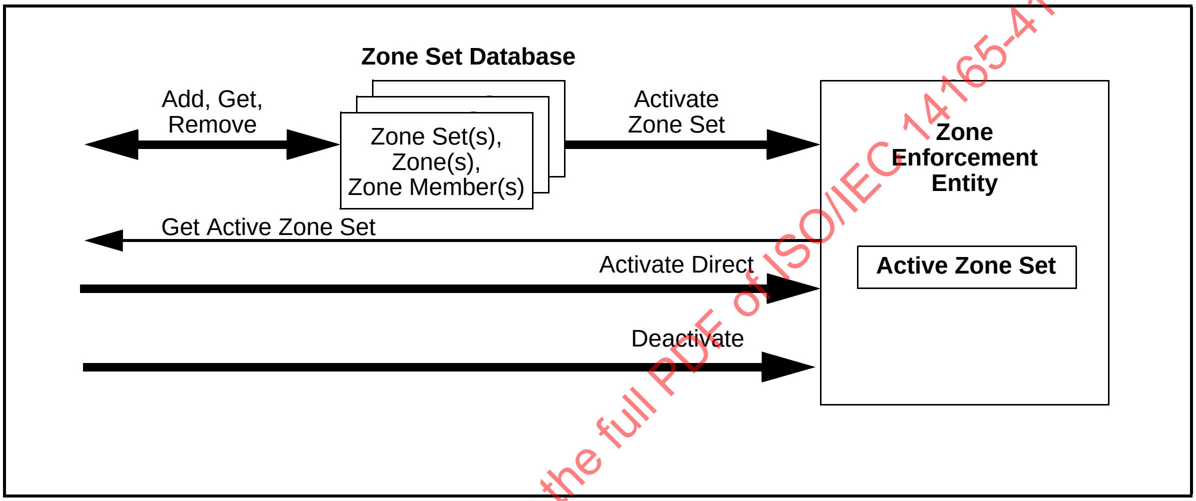


Figure 9 – Zone Set Database and Zone enforcement

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 ports may communicate without restriction, or no ports may communicate, depending on the implemented default zone behaviour.

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 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.

6.3.3 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.

Synchronous transactions shall be used, as defined in 4.5.2. 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 121. The Command Code shall be set as specified in 6.3.6.3 and 6.3.7.7.

6.3.4 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.

Table 236 – Control Zoning Management Requests

Code (hex)	**Mnemonic & Description**	**Attribute(s) in Request CT_IU**	**Attribute(s) in Accept CT_IU**
0142	GFEZ Get Fabric Enhanced Zoning Support	none	Fabric Enhanced Zoning support flags; List of Switch Enhanced Zoning support flags
0242	SFEZ Set Fabric Enhanced Zoning Support	Fabric Enhanced Zoning request flags	none

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 behaviour is depicted in figure 10.

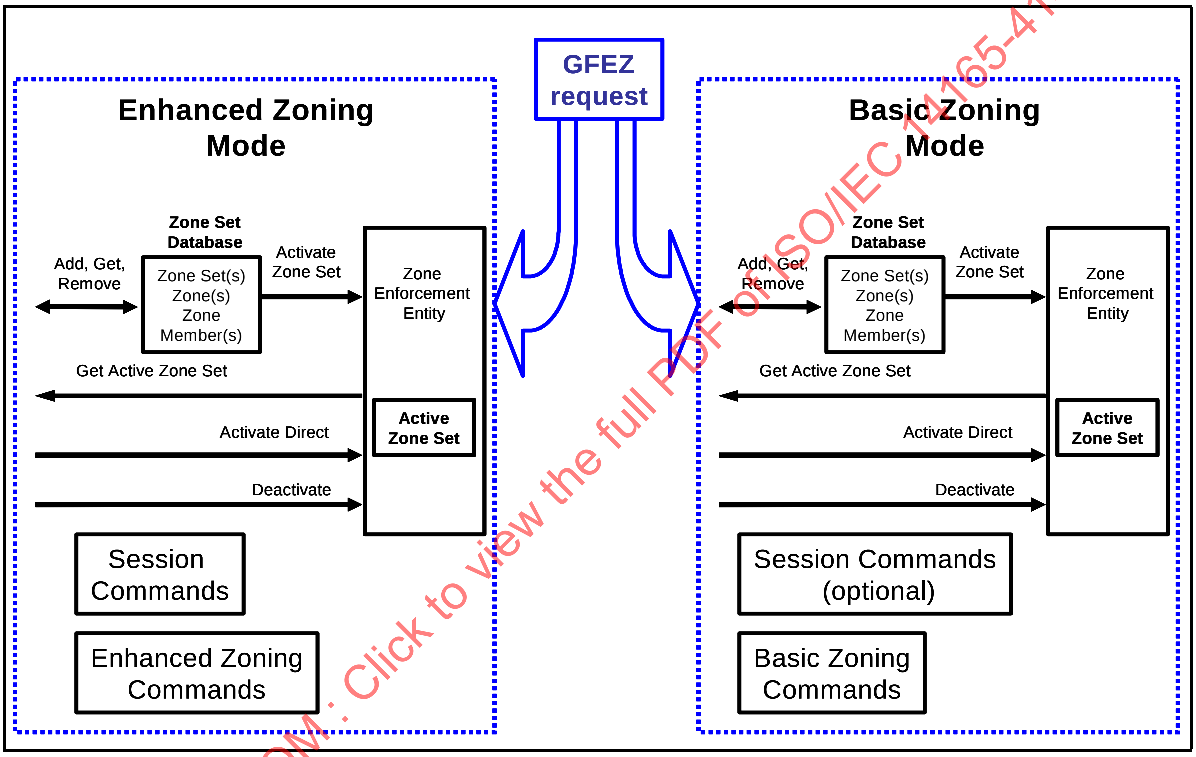


Figure 10 – Zoning Management model

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.

6.3.5 Default Zoning

For both Basic and Enhanced Zoning, there shall exist a Default Zone. The Default Zone has the following properties:

- any Nx_Port that is not a member of any zone in the active Zone Set shall be a member of the Default Zone;
- the fabric may not allow members of the Default Zone to interact with any other members in the Default Zone and
- the fabric shall not allow the members of the Default Zone to interact with members of any other zone.

6.3.6 Basic Zoning Management

6.3.6.1 Overview

In the context of Basic Zoning Management an interaction with the Fabric Zone Server may occur at any time, without any GS 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 6.4.6), Basic Zoning defines no Fabric behaviour 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 behaviour 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.

6.3.6.2 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.

6.3.6.3 Basic Zoning Management Commands

See table 237 for the CT requests defined to manage the Zoning Database in Basic Zoning Management context.

Table 237 – Fabric Zone Server – Basic Zoning Request Command Codes

Code (hex)	**Mnemonic & Description**	**Attribute(s) in Request CT_IU**	**Attribute(s) in Accept CT_IU**
0100	GZC Get Capabilities	none	Capabilities[a]
0111	GEST Get Enforcement State	none	Enforcement state[a]
0112	GZSN Get Zone Set List	none	List of Zone Set Name and Number of Zones
0113	GZD Get Zone List	Zone Set Name	List of Zone Names and Number of Zone Members
0114	GZM Get Zone Member List	Zone Name	List of Zone Member Identifier Types and Zone Member Identifiers
0115	GAZS Get Active Zone Set	none	Zone Set Name; Number of Zones; List of Zone Names, Number of Zone Members, List of Zone Member Identifier Types and Zone Member Identifiers

Table 237 – Fabric Zone Server – Basic Zoning Request Command Codes (Continued)

Code (hex)	**Mnemonic & Description**	**Attribute(s) in Request CT_IU**	**Attribute(s) in Accept CT_IU**
0116	GZS Get Zone Set	Zone Set Name	Zone Set Name; Number of Zones; List of Zone Names, Number of Zone Members, List of Zone Member Identifier Types and Zone Member Identifiers
0117	GAR Get Activation Results	none	see table 293
0200	ADZS Add Zone Set	Zone Set Name; Number of Zones; List of Zone Names, Number of Zone Members, List of Zone Member Identifier Types and Zone Member Identifiers	none
0201	AZSD Activate Zone Set Direct	Zone Set Name; Number of Zones; List of Zone Names, Number of Zone Members, List of Zone Member Identifier Types and Zone Member Identifiers	none
0202	AZS Activate Zone Set	Zone Set Name	none
0203	DZS Deactivate Zone Set	none	none
0204	AZM Add Zone Members	Zone Name; List of Zone Member Identifier Types and Zone Member Identifiers	none

Table 237 – Fabric Zone Server – Basic Zoning Request Command Codes (Continued)

Code (hex)	**Mnemonic & Description**	**Attribute(s) in Request CT_IU**	**Attribute(s) in Accept CT_IU**
0205	AZD Add Zone	Zone Set Name; Zone Name	none
0300	RZM Remove Zone Members	Zone Name; List of Zone Member Identifier Types and Zone Member Identifiers	none
0301	RZD Remove Zone	Zone Set Name; Zone Name	none
0302	RZS Remove Zone Set	Zone Set Name	none

[a] This is not an attribute.

6.3.7 Enhanced Zoning Management

6.3.7.1 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 Zoning session. This rule applies to both the Zone Set Database and the Active Zone Set. A Zoning session is delimited by the CT requests Server Session Begin (SSB) and Server Session End (SSE), directed to the Management Service and with GS_Subtype specifying the Zone Server. To ensure that Zoning Database information is reported consistently, Query requests to the Fabric Zone Server should be issued inside a Zoning session.

The Zoning session in Enhanced Zoning is associated with a lock on the Fabric (see FC-SW-3). 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.

6.3.7.2 Zone Types

6.3.7.2.1 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 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 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 a 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.

6.3.7.2.2 Broadcast Zoning

6.3.7.2.2.1 Overview

Broadcast zoning is enabled by using a broadcast attribute on zones that contain 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.

6.3.7.2.2.2 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 port

6.3.7.2.2.3 Zoning Not Active (i.e., no Active Zoneset)

Broadcast frames are delivered to all logged in Nx_Ports.

6.3.7.2.2.4 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 all the devices on the loop (see FC-AL-2).

6.3.7.3 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.

6.3.7.4 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.

6.3.7.5 Management Rules

Management of the Zone Set Database is subject to the following rules:

- 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;
- 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
- access to the management function is serialized by the Fabric using the session protocol described in FC-SW-3. 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.

6.3.7.6 Session Commands

See table 238 for the CT requests defined to manage the Zoning session in Enhanced Zoning Management context.

Table 238 – Fabric Zone Server – Session Request Command Codes

Code (hex)	**Mnemonic & Description**	**Attribute(s) in Request CT_IU**	**Attribute(s) in Accept CT_IU**
7FF9	SSB Server Session Begin[a]	none	none
7FFA	SSE Server Session End[b]	none	none
7FFB	CMIT Commit Zone Changes[c]	none	none

[a] Coupled with the ACA SW_ILS processing (see FC-SW-3).

[b] Coupled with the RCA SW_ILS processing (see FC-SW-3).

[c] Coupled with the SFC and UFC SW_ILSs processing (see FC-SW-3).

6.3.7.7 Enhanced Zoning Management Commands

See table 239 for the CT requests defined to manage the Zoning Database in Enhanced Zoning Management context.

Table 239 – Fabric Zone Server – Enhanced Zoning Request Command Codes

Code (hex)	**Mnemonic**	**Description**	**Request Attributes**	**Accept Attributes**
0117	GAR	Get Activation Results	none	see table 293
0120	GZA	Get Zone Attribute Object Name	Zone Name	Zone Attribute Object Name
0121	GZAB	Get Zone Attribute Block	Zone Attribute Object Name	Zone Attribute Block
0122	GZSE	Get Zone Set List - Enhanced	Null	List of Zone Set Names
0123	GZDE	Get Zone List - Enhanced	Zone Set Name	List of Zone Names
0124	GZME	Get Zone Member List - Enhanced	Zone Name	Zone Member List
0125	GZAL	Get Zone Attribute Object List	Null	Zone Attribute Object List
0128	GAL	Get Alias List	null	Alias List
0129	GAM	Get Alias Member List	Alias Name	Alias Member List

Table 239 – Fabric Zone Server – Enhanced Zoning Request Command Codes (Continued)

Code (hex)	Mnemonic	Description	Request Attributes	Accept Attributes
0227	SZA	Set Zone Attribute Object Name	Zone Name, Zone Attribute Object Name	Null
0228	SZAB	Set Zone Attribute Block	Zone Attribute Object Name, Zone Attribute Block	Null
0220	CZS	Create Zone Set	Zone Set Name, Zone Name List	Null
0225	CZ	Create Zone	Zone Name, Zone Member List	Null
0229	CA	Create Alias	Alias Name, Alias Member List	Null
0226	CZA	Create Zone Attribute Object	Zone Attribute Object Name, Zone Attribute Block	Null
0221	AZ	Add Zones	Zone Set Name, Zone Name List	Null
0321	RZ	Remove Zones	Zone Set Name, Zone Name List	Null
0224	AZME	Add Zone Members - Enhanced	Zone Name, Zone Member List	Null
0324	RZME	Remove Zone Members - Enhanced	Zone Name, Zone Member List	Null
022A	AAM	Add Alias Members	Alias Name, Alias Member List	Null

Table 239 – Fabric Zone Server – Enhanced Zoning Request Command Codes (Continued)

Code (hex)	Mnemonic	Description	Request Attributes	Accept Attributes
032A	RAM	Remove Alias Members	Alias Name, Alias Member List	Null
032B	DLZS	Delete Zone Set	Zone Set Name	Null
032C	DLZ	Delete Zone	Zone Name	Null
032D	DLA	Delete Alias	Alias Name	Null
032E	DLZA	Delete Zone Attribute Object	Zone Attribute Object Name	Null
0125	GAZSE	Get Active Zone Set - Enhanced	Null	Active Zone Set Object
0211	AZSDE	Activate Zone Set Direct - Enhanced[a]	Zone Set Object	Null
0212	AZSE	Activate Zone Set - Enhanced	Zone Set Name	Null
0213	DZSE	Deactivate Zone Set - Enhanced	Null	Null

[a] The Zone Set defined in the Request CT_IU is not added to the Zone Set Database.

6.3.8 Zoning Data Structures

6.3.8.1 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 240:

Table 240 – General Name Format

Item	Size bytes
Name Length (n)	1
Reserved	3
Name Value	n
Fill bytes (m)	0, 1, 2 or 3

6.3.8.1.1 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.

6.3.8.1.2 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 name shall be between 1 and 64 characters in length (n);
- all characters shall be 7 bit ASCII characters;
- 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
- 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 following symbols ($ to ^_).

6.3.8.1.3 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).

6.3.8.2 Basic Zoning Data Structures

6.3.8.2.1 Overview

In Basic Zoning framework the Fabric Zone Server object model is shown in figure 11.

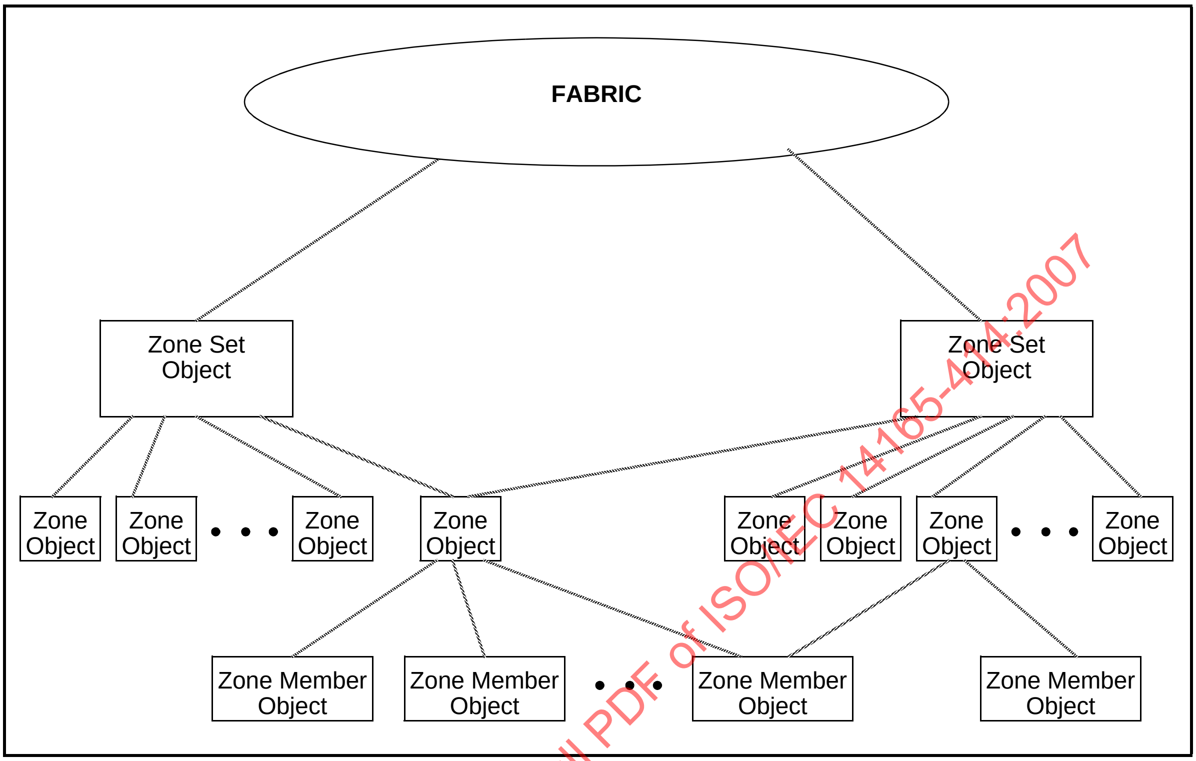


Figure 11 – Fabric Zone Server Object Model

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 12.

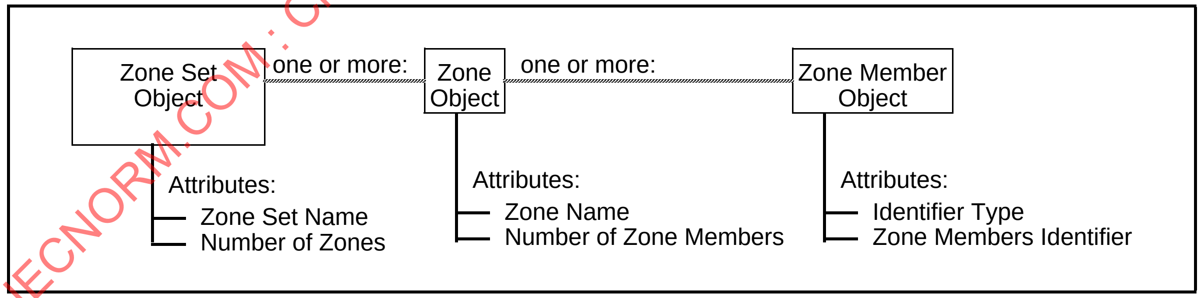


Figure 12 – Zone Set, Zone and Zone Member attributes

6.3.8.2.2 Zone Set Object

6.3.8.2.2.1 Zone Set Name

The format of the Zone Set Name attribute, as used by the Fabric Zone Server, shall follow the general name format described in 6.3.8.1.

6.3.8.2.2.2 Number of Zones

The format of the Number of Zones attribute, as used by the Fabric Zone Server, shall be as shown in table 241. This attribute shall contain the integer number of zones in the Zone Set.

Table 241 – Number of Zones Format

Item	**Size** bytes
Number of Zones	4

This attribute may be defined using the protocol described in 6.3.6.

6.3.8.2.3 Zone Object

6.3.8.2.3.1 Zone Name

The format of the Zone Name attribute, as used by the Fabric Zone Server, shall follow the general name format described in 6.3.8.1.

6.3.8.2.3.2 Number of Zone Members

The format of the Number of Zone Members attribute, as used by the Fabric Zone Server, shall be as shown in table 242. This attribute shall contain the integer number of members in the specified Zone.

Table 242 – Number of Zone Members Format

Item	**Size** bytes
Number of Zone Members	4

This attribute may be defined using the protocol described in 6.3.6. The null value for the Number of Zone Members attribute is 00 00 00 00h.

6.3.8.2.4 Zone Member Object

6.3.8.2.4.1 Zone Member Identifier Type

The format of the Zone Member Identifier Type attribute, as used by the Fabric Zone Server, shall be as shown in table 243. This attribute establishes the format of the information contained in the Zone Member Identifier attribute.

Table 243 – Zone Member Identifier Type–encoding

Encoded value (hex)	**Description**
01	N_Port_Name
02	Domain_ID and physical port
03	N_Port_ID
04	Node_Name
all others	Reserved

NOTE 1 The values specified in table 243 differ from those specified in FC-SW-3 for backward compatibility.

NOTE 2 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 behaviour 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 6.3.6.

NOTE 1 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 2 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 Port Indentifier associated with a Domain_ID and physical port number (see 5.2.5.33).

6.3.8.2.4.2 Zone Member Identifier

The format of the Zone Member Identifier attribute, as used by the Fabric Zone Server, 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 244.

Table 244 – 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 245.

Table 245 – 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 246.

Table 246 – 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 247.

Table 247 – Zone Member Identifier Format - Node_Name

Item	**Size** bytes
Node_Name	8

6.3.8.3 Enhanced Zoning Data Structures

6.3.8.3.1 Zone Set Database

The currently defined Zoning Object types are

- Zone Set Objects,
- Zone Objects,
- Zone Alias Objects and
- Zone Attribute Objects.

The Zoning Objects have an Object Identifier as defined in table 248.

Table 248 – 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 6.3.8.1.

Figure 13 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.

6.3.8.3.2 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 14.

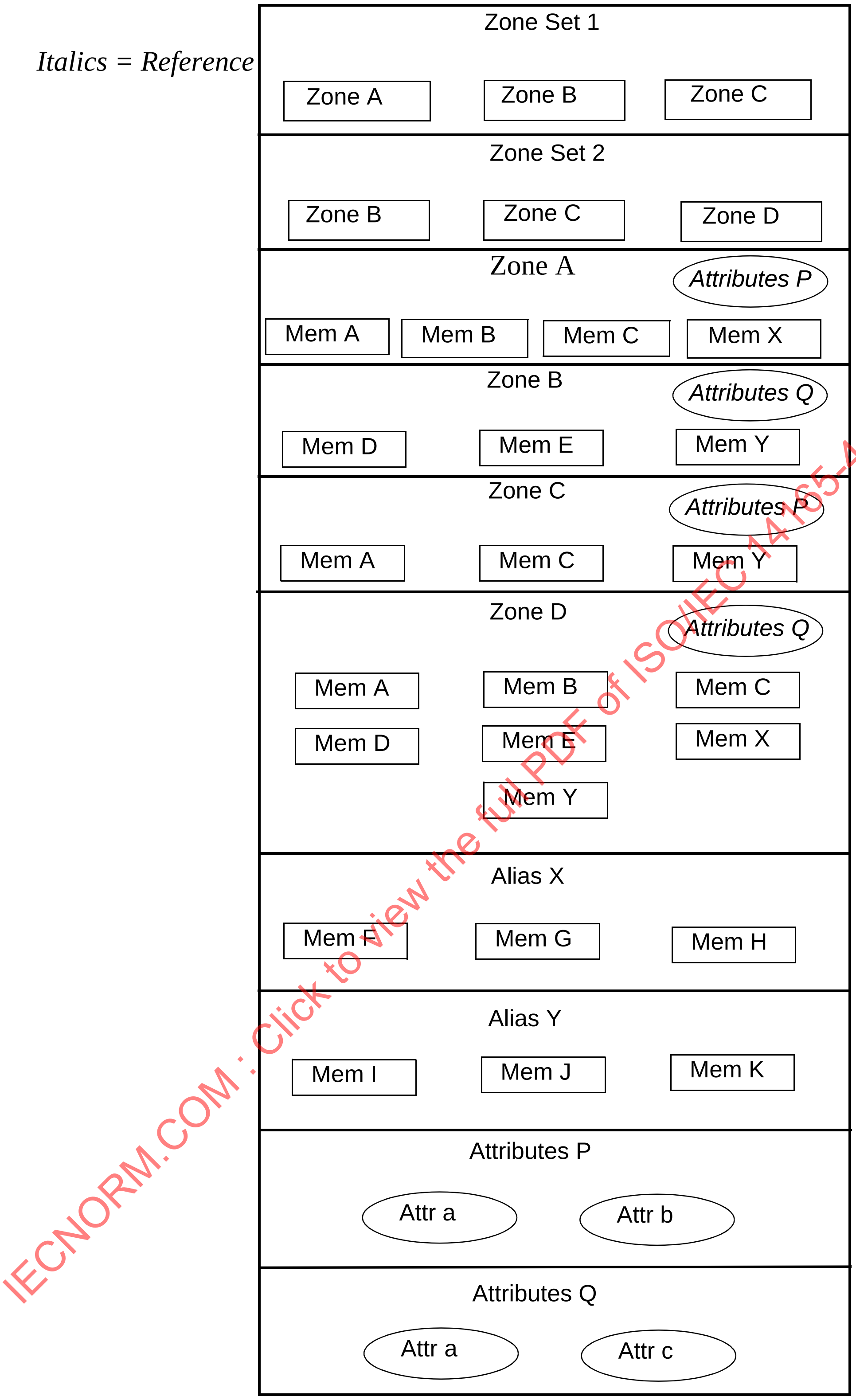


Figure 13 – Logical Structure of the Zone Set Database

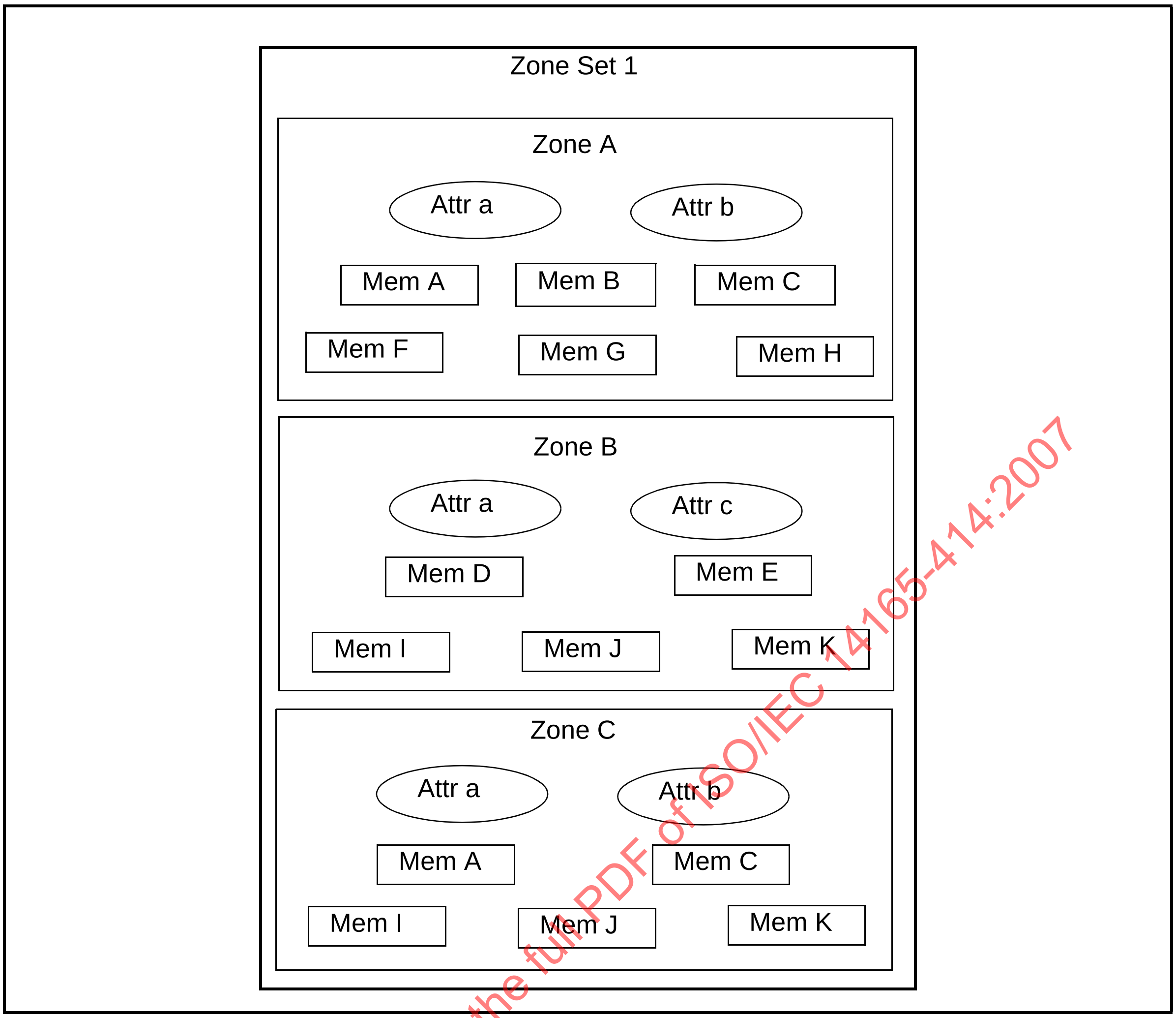


Figure 14 – Logical Structure of the Active Zone Set

6.3.8.3.3 Zone Set Object

6.3.8.3.3.1 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 249.

Table 249 – 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 6.3.8.1
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 6.3.8.1.

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.

6.3.8.3.3.2 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 250.

Table 250 – 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.

6.3.8.3.4 Zone Reference Object

The Zone Reference Object format is depicted in table 251.

Table 251 – Zone Reference Object

Item	**Value**	**Size** bytes
Object Identifier	04	1
Reserved	00	3
Zone Name	general name	see 6.3.8.1

Zone Name: The format of the Zone Name attribute shall follow the general name format described in 6.3.8.1.

6.3.8.3.5 Zone Object

6.3.8.3.5.1 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 252.

Table 252 – Zone Object in the Zone Set Database

Item	Value	Size bytes
Object Identifier	02	1
Reserved	00	3
Zone Name	general name	see 6.3.8.1
Zone Attribute Object Name	general name	see 6.3.8.1
Number of Zone Members	*n*	4
Zone Member #1	Zone Member object	see 6.3.8.3.6
Zone Member #2	Zone Member object	see 6.3.8.3.6
...	...	...
Zone Member #*n*	Zone Member object	see 6.3.8.3.6

Zone Name: The format of the Zone Name attribute shall follow the general name format described in 6.3.8.1.

Zone Attribute Object Name: The format of the Zone Attribute Object Name attribute shall follow the general name format described in 6.3.8.1. 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 6.3.8.3.6.

6.3.8.3.5.2 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 253.

Table 253 – Zone Object in the Active Zone Set

Item	**Value**	**Size** bytes
Object Identifier	02	1
Reserved	00	3
Zone Name	general name	see 6.3.8.1
Number of Zone Attribute Entries	m	4
Zone Attribute Entry #1	Zone Attribute Entry	see table 261
Zone Attribute Entry #2	Zone Attribute Entry	see table 261
...	...	...
Zone Attribute Entry #m	Zone Attribute Entry	see table 261
Number of Zone Members	*n*	4
Zone Member #1	Zone Member object	see 6.3.8.3.6
Zone Member #2	Zone Member object	see 6.3.8.3.6
...	...	...
Zone Member #*n*	Zone Member object	see 6.3.8.3.6

Zone Name: The format of the Zone Name attribute shall follow the general name format described in 6.3.8.1.

Number of Zone Attribute Entries: This attribute shall contain the integer number of Zone Attribute Entries in the Zone.

Zone Attribute Entry: See 6.3.8.3.8.

Number of Zone Members: This attribute shall contain the integer number of Zone Members in the Zone.

Zone Member: See 6.3.8.3.6.

6.3.8.3.6 Zone Member Object

The Zone Member Object has the format depicted in table 254 and the attributes described below.

Table 254 – Zone Member Object Format

Item	**Size** bytes
Zone Member Identifier Type	1
Reserved	3
Zone Member Identifier value	x

Zone Member Identifier Type: The format of the Zone Member Identifier Type attribute, shall be as shown in table 255. This attribute establishes the format of the information contained in the Zone Member Identifier attribute.

Table 255 – Zone Member Identifier Type–encoding

Encoded value (hex)	**Description**
01	N_Port_Name
02	Domain_ID and physical port
03	N_Port_ID
04	Node_Name
05	Alias Name
06	F_Port_Name
E0 to FF	Vendor-Specific
all others	Reserved

If the Fabric Zone Server receives a request containing some unsupported member types, that request shall be rejected.

NOTE 1 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 2 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 Port Indentifier associated with a Domain_ID and physical port number (see 5.2.5.33).

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 244.

The Domain_ID and Port Zone Member Identifier format shall be as shown in table 245.

The N_Port_ID Zone Member Identifier format shall be as shown in table 246.

The Node_Name Zone Member Identifier format shall be as shown in table 247.

The F_Port_Name Zone Member Identifier format shall be as shown in table 256.

Table 256 – 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 6.3.8.1, and it is referencing a Zone Alias object.

The Vendor-Specific Zone Member Identifier format is depicted in table 257.

Table 257 – 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 eight byte T10 administered Vendor Id.

Vendor Specific Value Length: This field contains the length, in bytes, of the Vendor Sepcific 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*).

6.3.8.3.7 Zone Alias Object

The Zone Alias Object has the attributes described below and the format depicted in table 258.

Table 258 – Zone Alias Object

Item	**Value**	**Size** bytes
Object Identifier	03	1
Reserved	00	3
Zone Alias Name	general name	see 6.3.8.1.2
Number of Zone Alias Members	*n*	4
Zone Alias Member #1	Zone Member object	see 6.3.8.3.6
Zone Alias Member #2	Zone Member object	see 6.3.8.3.6
...	...	...
Zone Alias Member #n	Zone Member object	see 6.3.8.3.6

Zone Alias Name: The format of the Zone Alias Name attribute shall follow the general name format described in 6.3.8.1.

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 6.3.8.3.6. All Zone Member Identifier Types may be used, with the exception of the Alias Name member (type '05').

6.3.8.3.8 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 259.

Table 259 – Zone Attribute Object

Item	**Value**	**Size** bytes
Object Identifier	05	1
Reserved	00	3
Zone Attribute Object Name	general name	see 6.3.8.1.2
Zone Attribute Block	Zone Attribute Block (see table 260)	w

Table 260 – 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 6.3.8.1.

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 261.

Table 261 – 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 262 and shall be restricted to a value between 0000h and 00FFh.

Table 262 – Zone Attribute Types

Zone Attribute Type (hex)	**Description**	**Zone Attribute Required**
0001	Protocol[a]	Yes
0002	Hard	None
0003	Broadcast	None
00E0	Vendor Specific	Yes
other values	Reserved	

[a] For a definition of Protocol attribute type, see FC-SW-3.

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:

- Zone Attribute Type,
- Zone Attribute Length and
- 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 type Zone Attribute Type field (see table 262). When a Protocol Attribute Type is specified, the format of the Protocol Attribute Value is depicted in table 263. When a Vendor-Specific Attribute Type is specified, the format of the Vendor-Specific Attribute Value is depicted in table 264.

Table 263 – Protocol Attribute Value

Item	**Size** bytes
FC-4 Type	1
Reserved	3

FC-4 Type: The FC-4 type (see FC-FS) 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.

Table 264 – Vendor-Specific Attribute Value

Item	**Size** bytes
Vendor Identifier	8
Vendor-Specific Value	*n*
Pad	*m*

Vendor Identifier: Contains the eight byte T10 administered Vendor Id.

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*).

6.3.9 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 265, is returned.

Table 265 – Reject CT_IU Reason code explanations

Encoded value (hex)	**Description**
00	No additional explanation
01	Zones not supported
10	Zone Set Name unknown
11	No Zone Set active
12	Zone Name unknown
13	Zone State unknown
14	Incorrect payload length
15	Zone Set to be activated too large
16	Deactivate Zone Set failed
17	Request not supported
18	Capability not supported
19	Zone Member Identifier Type not supported
1A	Invalid Zone Set definition
20	Enhanced Zoning Commands Not Supported
21	Zone Set Already Exists
22	Zone Already Exists
23	Alias Already Exists
24	Zone Set Unknown
25	Zone Unknown
26	Alias Unknown
28	Unknown Zone Attribute Type

Table 265 – Reject CT_IU Reason code explanations (Continued)

Encoded value (hex)	**Description**
29	Fabric Unable to Work in Enhanced Mode
30	Basic Zoning Commands Not Supported
31	Zone Attribute Object Already Exists
32	Zone Attribute Object Unknown
33	Request in Process
34	CMIT in Process
35	Hard Enforcement Failed
36	Unresolved references in the Zone Set Database
37	Consistency checks failed
Others	Reserved

6.3.10 Zoning Management Commands

6.3.10.1 Control Zoning Commands

6.3.10.1.1 Get Fabric Enhanced Zoning Support (GFEZ)

The Fabric Zone Server shall, when 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 266.

Table 266 – 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 267.

Table 267 – 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 268.

Table 268 – 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 269.

Table 269 – 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 to 31	Reserved

The Switch Enhanced Zoning support flags are shown in table 270.

Table 270 – 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 to 31	Reserved

6.3.10.1.2 Set Fabric Enhanced Zoning Support (SFEZ)

The Fabric Zone Server shall, when 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 GS 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 271.

Table 271 – 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 272.

Table 272 – 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 to 31	Reserved

The format of the Accept CT_IU to a SFEZ request is shown in table 273.

Table 273 – Accept CT_IU to SFEZ Request

Item	**Size** bytes
CT_IU preamble	see 4.3

6.3.10.2 Session Commands

6.3.10.2.1 Server Session Begin (SSB)

See 4.9.5.2.

6.3.10.2.2 Server Session End (SSE)

See 4.9.5.3.

6.3.10.2.3 Commit Zone Changes (CMIT)

The Fabric Zone Server shall, when 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 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 274 below.

Table 274 – CMIT Request Payload

Item	**Size** bytes
CT_IU preamble	see 4.3

The format of the CMIT Accept payload is depicted in table 275 below:

Table 275 – CMIT Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3

6.3.10.3 Basic Zoning Commands

6.3.10.3.1 Query – Get Capabilities (GZC)

The Fabric Zone Server shall, when 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 276.

Table 276 – 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 277.

Table 277 – 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 278.

Table 278 – 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 6.3.10.3.10). 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.

6.3.10.3.2 Query – Get Enforcement State (GEST)

The Fabric Zone Server shall, when 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 279.

Table 279 – 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 280.

Table 280 – 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 281.

Table 281 – 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.

6.3.10.3.3 Query – Get Zone Set List (GZSN)

The Fabric Zone Server shall, when 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 282.

Table 282 – 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 283.

Table 283 – 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 6.3.8.1
Number of Zones #1	4
Zone Set Name #2	see 6.3.8.1
Number of Zones #2	4
...	
Zone Set Name #*n*	see 6.3.8.1
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.

6.3.10.3.4 Query – Get Zone List (GZD)

The Fabric Zone Server shall, when 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 284.

Table 284 – 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 285.

Table 285 – 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 6.3.8.1
Number of Zone Members #1	4
Zone Name #2	see 6.3.8.1
Number of Zone Members #2	4
...	
Zone Name #*n*	see 6.3.8.1
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.

6.3.10.3.5 Query – Get Zone Member List (GZM)

The Fabric Zone Server shall, when 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 286.

Table 286 – GZM Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
Zone Name	see 6.3.8.1

The format of the Accept CT_IU to a GZM request is shown in table 287.

Table 287 – 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.

6.3.10.3.6 Query – Get Active Zone Set (GAZS)

The Fabric Zone Server shall, when 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 288.

Table 288 – 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 289.

Table 289 – Accept CT_IU to GAZS Request

Item	**Size** bytes
CT_IU preamble	see 4.3
Zone Set Name of Active Zone Set	see 6.3.8.1
Number of Zones	4
Zone Name #1	see 6.3.8.1
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 6.3.8.1

Table 289 – 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 6.3.8.1
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

6.3.10.3.7 Query – Get Zone Set (GZS)

The Fabric Zone Server shall, when 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 290.

Table 290 – GZS Request CT_IU

Item	Size bytes
CT_IU preamble	see 4.3
Zone Set Name	see 6.3.8.1

The format of the Accept CT_IU to a GZS request is shown in table 291.

Table 291 – Accept CT_IU to GZS Request

Item	Size bytes
CT_IU preamble	see 4.3
Zone Set Name of Zone Set	see 6.3.8.1
Number of Zones	4
Zone Name #1	see 6.3.8.1
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 291 – Accept CT_IU to GZS Request (Continued)

Item	**Size** bytes
Zone Name #2	see 6.3.8.1
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
...	

Table 291 – Accept CT_IU to GZS Request (Continued)

Item	Size bytes
Zone Name #*n*	see 6.3.8.1
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

6.3.10.3.8 Get Activation Results (GAR)

The Fabric Zone Server shall, when 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 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 292.

Table 292 – 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 293.

Table 293 – 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 Type #1 Domain ID	1
Reserved	3
Interconnect Element #1 Reason Code	1
Interconnect Element Name #2	8
Reserved	3
Interconnect Element Type #2 Domain ID	1
Reserved	3
Interconnect Element #2 Reason Code	1

Interconnect Element Name #*n*	8
Reserved	3
Interconnect Element Type #*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-3.

6.3.10.3.9 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 294.

Table 294 – ADZS Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
Zone Set Name	see 6.3.8.1
Number of Zones	4
Zone Name #1	see 6.3.8.1
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 6.3.8.1
Number of Zone Members #2	4
Zone Member Identifier Type #1	1
Reserved	3
Zone Member Identifier #1	8

Table 294 – 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 6.3.8.1
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 295.

Table 295 – ADZS Accept CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3

6.3.10.3.10 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 296.

Table 296 – AZSD Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
Zone Set Name	see 6.3.8.1
Number of Zones	4
Zone Name #1	see 6.3.8.1
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 6.3.8.1
Number of Zone Members #2	4
Zone Member Identifier Type #1	1
Reserved	3

Table 296 – 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 6.3.8.1
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 297.

Table 297 – AZSD Accept CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3

6.3.10.3.11 Activate Zone Set (AZS)

The Fabric Zone Server shall, when 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 298.

Table 298 – AZS Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
Zone Set Name to Activate	see 6.3.8.1

The format of the Accept CT_IU to an AZS request is shown in table 299.

Table 299 – Accept CT_IU to AZS Request

Item	**Size** bytes
CT_IU preamble	see 4.3

6.3.10.3.12 Deactivate Zone Set (DZS)

The Fabric Zone Server shall, when it receives a DZS request, deactivate the current Active Zone Set. The format of the DZS Request CT_IU is shown in table 300.

Table 300 – 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 301.

Table 301 – Accept CT_IU to DZS Request

Item	**Size** bytes
CT_IU preamble	see 4.3

6.3.10.3.13 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 302.

Table 302 – AZM Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
Zone Name	see 6.3.8.1
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 303.

Table 303 – AZM Accept CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3

6.3.10.3.14 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 304.

Table 304 – AZD Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
Zone Set Name	see 6.3.8.1
Zone Name	see 6.3.8.1

The format of the AZD Accept CT_IU is shown in table 305.

Table 305 – AZD Accept CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3

6.3.10.3.15 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 shown in table 306.

Table 306 – RZM Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
Zone Name	see 6.3.8.1
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 307.

Table 307 – RZM Accept CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3

6.3.10.3.16 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 308.

Table 308 – RZD Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
Zone Set Name	see 6.3.8.1
Zone Name	see 6.3.8.1

The format of the RZD Accept CT_IU is shown in table 309.

Table 309 – RZD Accept CT_IU

Item	Size bytes
CT_IU preamble	see 4.3

6.3.10.3.17 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 310.

Table 310 – RZS Request CT_IU

Item	Size bytes
CT_IU preamble	see 4.3
Zone Set Name	see 6.3.8.1

The format of the RZS Accept CT_IU is shown in table 311.

Table 311 – RZS Accept CT_IU

Item	Size bytes
CT_IU preamble	see 4.3

6.3.10.4 Enhanced Zoning Commands

6.3.10.4.1 Get Activation Results (GAR)

See 6.3.10.3.8.

6.3.10.4.2 Get Zone Attribute Object Name (GZA) Operation

The Fabric Zone Server shall, when 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 312 below.

Table 312 – GZA Request Payload

Item	**Size** bytes
CT_IU preamble	see 4.3
Zone Name	see 6.3.8.1

The format of the GZA accept payload is depicted in table 313 below

Table 313 – GZA Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3
Zone Attribute Object Name	see 6.3.8.1

6.3.10.4.3 Get Zone Attribute Block (GZAB) Operation

The Fabric Zone Server shall, when 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 314 below.

Table 314 – GZAB Request Payload

Item	Size bytes
CT_IU preamble	see 4.3
Zone Attribute Object Name	see 6.3.8.1

The format of the GZAB accept payload is depicted in table 315 below

Table 315 – GZAB Accept Payload

Item	Size bytes
CT_IU preamble	see 4.3
Zone Attribute Block	see 6.3.8.3.8

The format of the Zone Attribute Block is described in 6.3.8.3.8.

6.3.10.4.4 Get Zone Set List - Enhanced (GZSE) Operation

The Fabric Zone Server shall, when 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 316 below.

Table 316 – GZSE Request Payload

Item	Size bytes
CT_IU preamble	see 4.3

The format of the GZSE accept payload is depicted in table 317 below.

Table 317 – GZSE Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3
Number of Zone Set Names (*n*)	4
Zone Set Name 1	see 6.3.8.1
Zone Set Name 2	see 6.3.8.1
...	
Zone Set Name n	see 6.3.8.1

6.3.10.4.5 Get Zone List - Enhanced (GZDE) Operation

The Fabric Zone Server shall, when 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 an additional reason explanation of “Zone Set Unknown”.

The format of the GZDE request payload is shown in table 318 below.

Table 318 – GZDE Request Payload

Item	**Size** bytes
CT_IU preamble	see 4.3
Zone Set Name	see 6.3.8.1

The format of the GZDE accept payload is depicted in table 319 below.

Table 319 – GZDE Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3
Number of Zone Names (*n*)	4
Zone Name 1	see 6.3.8.1
Zone Name 2	see 6.3.8.1
...	
Zone Name *n*	see 6.3.8.1

6.3.10.4.6 Get Zone Member List - Enhanced (GZME) Operation

The Fabric Zone Server shall, when 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 320 below.

Table 320 – GZME Request Payload

Item	**Size** bytes
CT_IU preamble	see 4.3
Zone Name	see 6.3.8.1

The format of the GZME accept payload is depicted in table 321 below:

Table 321 – GZME Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3
Number of Zone Member Entries (*n*)	4
Zone Member Entry 1	see 6.3.8.2.4
Zone Member Entry 2	see 6.3.8.2.4
...	
Zone Member Entry *n*	see 6.3.8.2.4

The format of the Zone Member Entry is depicted in table 254.

6.3.10.4.7 Get Zone Attribute Object List (GZAL) Operation

The Fabric Zone Server shall, when 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 322 below.

Table 322 – GZAL Request Payload

Item	**Size** bytes
CT_IU preamble	see 4.3

The format of the GZAL accept payload is depicted in table 323 below.

Table 323 – 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 6.3.8.1
Zone Attribute Object Name 2	see 6.3.8.1
...	
Zone Attribute Object Name *n*	see 6.3.8.1

6.3.10.4.8 Get Alias List (GAL) Operation

The Fabric Zone Server shall, when 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 324 below:

Table 324 – GAL Request Payload

Item	**Size** bytes
CT_IU preamble	see 4.3

The format of the GAL Accept payload is depicted in table 325 below:

Table 325 – GAL Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3
Number of Alias names (n)	4
Alias Name 1	see 6.3.8.1
Alias Name 2	see 6.3.8.1
...	
Alias Name *n*	see 6.3.8.1

The format of the Alias Name is described in 6.3.8.1.

6.3.10.4.9 Get Alias Member List (GAM) Operation

The Fabric Zone Server shall, when 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 326 below:

Table 326 – GAM Request Payload

Item	**Size** bytes
CT_IU preamble	see 4.3
Alias Name	see 6.3.8.1

The format of the Alias Name is described in 6.3.8.1.

The format of the GAM Accept payload is depicted in table 327 below:

Table 327 – GAM Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3
Number of Alias Member Entries (*n*)	4
Alias Member Entry 1	see 6.3.8.3.7
Alias Member Entry 2	see 6.3.8.3.7
...	
Alias Member Entry *n*	see 6.3.8.3.7

The format of the Alias Member Entry is depicted in table 254.

6.3.10.4.10 Set Zone Attribute Object Name (SZA) Operation

The Fabric Zone Server shall, when 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 328 below.

Table 328 – SZA Request Payload

Item	**Size** bytes
CT_IU preamble	see 4.3
Zone Name	see 6.3.8.1
Zone Attribute Object Name	see 6.3.8.1

The format of the SZA Accept payload is depicted in table 329 below

Table 329 – SZA Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3

6.3.10.4.11 Set Zone Attribute Block (SZAB) Operation

The Fabric Zone Server shall, when 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 330 below.

Table 330 – SZAB Request Payload

Item	**Size** bytes
CT_IU preamble	see 4.3
Zone Attribute Object Name	see 6.3.8.1
Zone Attribute Block	see 6.3.8.3.8

The format of the Zone Attribute Block is described in 6.3.8.3.8.

The format of the SZAB Accept payload is depicted in table 331 below

Table 331 – SZAB Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3

6.3.10.4.12 Create Zone Set (CZS) Operation

The Fabric Zone Server shall, when 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 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 332 below.

Table 332 – CZS Request Payload

Item	**Size** bytes
CT_IU preamble	see 4.3
Zone Set Name	see 6.3.8.1
Number of Zone Names (*n*)	4
Zone Name 1	see 6.3.8.1
Zone Name 2	see 6.3.8.1
...	
Zone Name *n*	see 6.3.8.1

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 333 below:

Table 333 – CZS Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3

6.3.10.4.13 Create Zone (CZ) Operation

The Fabric Zone Server shall, when 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 334 below.

Table 334 – CZ Request Payload

Item	**Size** bytes
CT_IU preamble	see 4.3
Zone Name	see 6.3.8.1
Number of Zone Members (*n*)	4
Zone Member Entry 1	see 6.3.8.2.4
Zone Member Entry 2	see 6.3.8.2.4
...	
Zone Member Entry *n*	see 6.3.8.2.4

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 format of the Zone Member Entry is described in table 254.

The format of the CZ Accept payload is depicted in table 335 below:

Table 335 – CZ Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3

6.3.10.4.14 Create Alias (CA) Operation

The Fabric Zone Server shall, when 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 336 below.

Table 336 – CA Request Payload

Item	Size bytes
CT_IU preamble	see 4.3
Alias Name	see 6.3.8.1
Number of Alias Members (n)	4
Alias Member Entry 1	see 6.3.8.3.7
Alias Member Entry 2	see 6.3.8.3.7
...	
Alias Member Entry n	see 6.3.8.3.7

Alias Name: The format of the Alias Name is described in 6.3.8.1.

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 format of the Alias Member Entry is described in table 254.

The format of the CA Accept payload is depicted in table 337 below:

Table 337 – CA Accept Payload

Item	Size bytes
CT_IU preamble	see 4.3

6.3.10.4.15 Create Zone Attribute Object (CZA) Operation

The Fabric Zone Server shall, when 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 338 below.

Table 338 – CZA Request Payload

Item	Size bytes
CT_IU preamble	see 4.3
Zone Attribute Object Name	see 6.3.8.1
Zone Attribute Block	see 6.3.8.3.8

Zone Attribute Object Name: The format of the Zone Attribute Object Name is described in 6.3.8.1.

Zone Attribute Block: The format of the Zone Attribute Block is described in 6.3.8.3.8.

The format of the CA Accept payload is depicted in table 339 below:

Table 339 – CZA Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3

6.3.10.4.16 Add Zones Enhanced (AZ) Operation

The Fabric Zone Server shall, when 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 340 below.

Table 340 – AZ Request Payload

Item	**Size** bytes
CT_IU preamble	see 4.3
Zone Set Name	see 6.3.8.1
Number of Zone Names (*n*)	4
Zone Name 1	see 6.3.8.1
Zone Name 2	see 6.3.8.1
...	
Zone Name *n*	see 6.3.8.1

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 341 below:

Table 341 – AZ Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3

6.3.10.4.17 Remove Zones (RZ) Operation

The Fabric Zone Server shall, when 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 342 below.

Table 342 – RZ Request Payload

Item	**Size** bytes
CT_IU preamble	see 4.3
Zone Set Name	see 6.3.8.1
Number of Zone Names (*n*)	4
Zone Name 1	see 6.3.8.1
Zone Name 2	see 6.3.8.1
...	
Zone Name *n*	see 6.3.8.1

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 343 below:

Table 343 – RZ Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3

6.3.10.4.18 Add Zone Members - Enhanced (AZME) Operation

The Fabric Zone Server shall, when 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 344 below.

Table 344 – AZME Request Payload

Item	**Size** bytes
CT_IU preamble	see 4.3
Zone Name	see 6.3.8.1
Number of Zone Member Entries (*n*)	4
Zone Member Entry 1	see 6.3.8.2.4
Zone Member Entry 2	see 6.3.8.2.4
...	
Zone Member Entry *n*	see 6.3.8.2.4

The format of the Zone Member Entry is described in table 254.

The format of the AZME Accept payload is depicted in table 345 below:

Table 345 – AZME Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3

6.3.10.4.19 Remove Zone Members - Enhanced (RZME) Operation

The Fabric Zone Server shall, when 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 346 below.

Table 346 – RZME Request Payload

Item	**Size** bytes
CT_IU preamble	see 4.3
Zone Name	see 6.3.8.1
Number of Zone Member Entries (*n*)	4
Zone Member Entry 1	see 6.3.8.2.4
Zone Member Entry 2	see 6.3.8.2.4
...	
Zone Member Entry *n*	see 6.3.8.2.4

The format of the Zone Member Entry is described in table 254.

The format of the RZME Accept payload is depicted in table 347 below:

Table 347 – RZME Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3

6.3.10.4.20 Add Alias Members (AAM) Operation

The Fabric Zone Server shall, when 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 348 below.

Table 348 – AAM Request Payload

Item	**Size** bytes
CT_IU preamble	see 4.3
Alias Name	see 6.3.8.1
Number of Alias Member Entries (*n*)	4
Alias Member Entry 1	see 6.3.8.3.7
Alias Member Entry 2	see 6.3.8.3.7
...	
Alias Member Entry *n*	see 6.3.8.3.7

The format of the Alias Name is described in 6.3.8.1. Also, the format of the Alias Member Entry is described in table 254.

The format of the AAM Accept payload is depicted in table 349 below

Table 349 – AAM Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3

6.3.10.4.21 Remove Alias Members (RAM) Operation

The Fabric Zone Server shall, when 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 350 below.

Table 350 – RAM Request Payload

Item	**Size** bytes
CT_IU preamble	see 4.3
Alias Name	see 6.3.8.1
Number of Alias Member Entries (*n*)	4
Alias Member Entry 1	see 6.3.8.3.7
Alias Member Entry 2	see 6.3.8.3.7
...	
Alias Member Entry *n*	see 6.3.8.3.7

Alias Name: The format of the Alias Name is described in 6.3.8.1.

Alias Member Entry: The format of the Alias Member Entry is described in table 254.

The format of the RAM Accept payload is depicted in table 351 below.

Table 351 – RAM Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3

6.3.10.4.22 Delete Zone Set (DLZS) Operation

The Fabric Zone Server shall, when 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 352 below.

Table 352 – DLZS Request Payload

Item	**Size** bytes
CT_IU preamble	see 4.3
Zone Set Name	see 6.3.8.1

The format of the DLZS Accept payload is depicted in table 353 below:

Table 353 – DLZS Accept Payload

Item	Size bytes
CT_IU preamble	see 4.3

6.3.10.4.23 Delete Zone (DLZ) Operation

The Fabric Zone Server shall, when 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 354 below.

Table 354 – DLZ Request Payload

Item	Size bytes
CT_IU preamble	see 4.3
Zone Name	see 6.3.8.1

The format of the DLZ Accept payload is depicted in table 355 below:

Table 355 – DLZS Accept Payload

Item	Size bytes
CT_IU preamble	see 4.3

6.3.10.4.24 Delete Alias (DLA) Operation

The Fabric Zone Server shall, when 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 356 below.

Table 356 – DLA Request Payload

Item	**Size** bytes
CT_IU preamble	see 4.3
Alias Name	see 6.3.8.1

Alias Name: The format of the Alias Name is described in 6.3.8.1.

The format of the DLA Accept payload is depicted in table 357 below:

Table 357 – DLA Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3

6.3.10.4.25 Delete Zone Attribute Object (DLZA) Operation

The Fabric Zone Server shall, when 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 358 below.

Table 358 – DLZA Request Payload

Item	**Size** bytes
CT_IU preamble	see 4.3
Zone Attribute Object Name	see 6.3.8.1

Zone Attribute Object Name: The format of the Zone Attribute Object Name is described in 6.3.8.1.

The format of the DLA Accept payload is depicted in table 359 below:

Table 359 – DLZA Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3

6.3.10.4.26 Get Active Zone Set - Enhanced (GAZSE) Operation

The Fabric Zone Server shall, when 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 360.

Table 360 – 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 361.

Table 361 – Accept CT_IU to GAZSE Request

Item	**Size** bytes
CT_IU preamble	see 4.3
Number of Zones	4
Zone Name #1	see 6.3.8.1
Number of Zone Attribute Entries #1	4
Zone Attribute Entry #1	see table 261
Zone Attribute Entry #2	see table 261
...	
Zone Attribute Entry #*k*	see table 261
Number of Zone Members #1	4
Zone Member #1	see 6.3.8.3.6
Zone Member #2	see 6.3.8.3.6
...	
Zone Member #*m*	see 6.3.8.3.6
Zone Name #2	see 6.3.8.1
Number of Zone Attribute Entries #2	4
Zone Attribute Entry #1	see table 261
Zone Attribute Entry #2	see table 261
...	

Table 361 – Accept CT_IU to GAZSE Request (Continued)

Item	**Size** bytes
Zone Attribute Entry #k	see table 261
Number of Zone Members #2	4
Zone Member #1	see 6.3.8.3.6
Zone Member #2	see 6.3.8.3.6
...	
Zone Member #*m*	see 6.3.8.3.6
...	
Zone Name #*n*	see 6.3.8.1
Number of Zone Attribute Entries #*n*	4
Zone Attribute Entry #1	see table 261
Zone Attribute Entry #2	see table 261
...	
Zone Attribute Entry #k	see table 261
Number of Zone Members #*n*	4
Zone Member #1	see 6.3.8.3.6
Zone Member #2	see 6.3.8.3.6
...	
Zone Member #*m*	see 6.3.8.3.6

The format of the Zone Attribute Entry is defined in 6.3.8.3.8.

The format of the Zone Member is defined in 6.3.8.3.6.

6.3.10.4.27 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 362.

Table 362 – AZSDE Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
Number of Zones	4
Zone Name #1	see 6.3.8.1
Number of Zone Attribute Entries #1	4
Zone Attribute Entry #1	see table 261
Zone Attribute Entry #2	see table 261
...	
Zone Attribute Entry #k	see table 261
Number of Zone Members #1	4
Zone Member #1	see 6.3.8.3.6
Zone Member #2	see 6.3.8.3.6
...	
Zone Member #*m*	see 6.3.8.3.6
Zone Name #2	see 6.3.8.1
Number of Zone Attribute Entries #2	4
Zone Attribute Entry #1	see table 261
Zone Attribute Entry #2	see table 261
...	
Zone Attribute Entry #*k*	see table 261
Number of Zone Members #2	4
Zone Member #1	see 6.3.8.3.6
Zone Member #2	see 6.3.8.3.6
...	
Zone Member #*m*	see 6.3.8.3.6

Table 362 – AZSDE Request CT_IU (Continued)

Item	**Size** bytes
...	
Zone Name #*n*	see 6.3.8.1
Number of Zone Attribute Entries #*n*	4
Zone Attribute Entry #1	see table 261
Zone Attribute Entry #2	see table 261
...	
Zone Attribute Entry #*k*	see table 261
Number of Zone Members #*n*	4
Zone Member #1	see 6.3.8.3.6
Zone Member #2	see 6.3.8.3.6
...	
Zone Member #*m*	see 6.3.8.3.6

The format of the Zone Attribute Entry is defined in 6.3.8.3.8.

The format of the Zone Member is defined in 6.3.8.3.6.

The format of the AZSD Accept CT_IU is shown in table 363.

Table 363 – AZSDE Accept CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3

6.3.10.4.28 Activate Zone Set - Enhanced (AZSE) Operation

The Fabric Zone Server shall, when 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 364.

Table 364 – AZSE Request CT_IU

Item	**Size** bytes
CT_IU preamble	see 4.3
Zone Set Name to Activate	see 6.3.8.1

The format of the Accept CT_IU to an AZSE request is shown in table 365.

Table 365 – Accept CT_IU to AZSE Request

Item	**Size** bytes
CT_IU preamble	see 4.3

6.3.10.4.29 Deactivate Zone Set - Enhanced (DZSE) Operation

The Fabric Zone Server shall, when it receives a DZSE request, deactivate the current Active Zone Set. The format of the DZSE Request CT_IU is shown in table 366.

Table 366 – 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 367.

Table 367 – Accept CT_IU to DZSE Request

Item	**Size** bytes
CT_IU preamble	see 4.3

6.4 Security Policy Server

6.4.1 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 121.

Security Policy Server requests are carried over the GS Common Transport (FC-CT).

6.4.2 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.

Synchronous transactions shall be used, as defined in 4.5.2. 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.

For a complete command list and additional details, see FC-SP.

6.5 Fabric Device Management Interface

6.5.1 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 and FC-HBA).

6.5.2 FDMI Relationship to the Name Server

The FDMI supplements the services provided by the Name Server. To obtain complete information as provided by the HBA API, it is necessary for the client to obtain device information from both the Name Server and HBA Server.

6.5.3 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 368.

Table 368 – Fabric-Device Management Interface GS_Subtypes

Encoded value (hex)	**Description**
10	HBA Management Server
11 to 1F	Reserved for future use
others	see table 121

6.5.4 HBA Management Server

6.5.4.1 Overview

The HBA Management Server is defined to allow the registration and retrieval of HBA Management information.

6.5.4.2 Platform Model

The support of HBA management through the Fabric is based on the Platform model shown in figure 15. Specifically, a Platform contains one or more Nodes and each Node contains one or more Ports. An HBA shall be associated with one and only one Node. However, 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.

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.

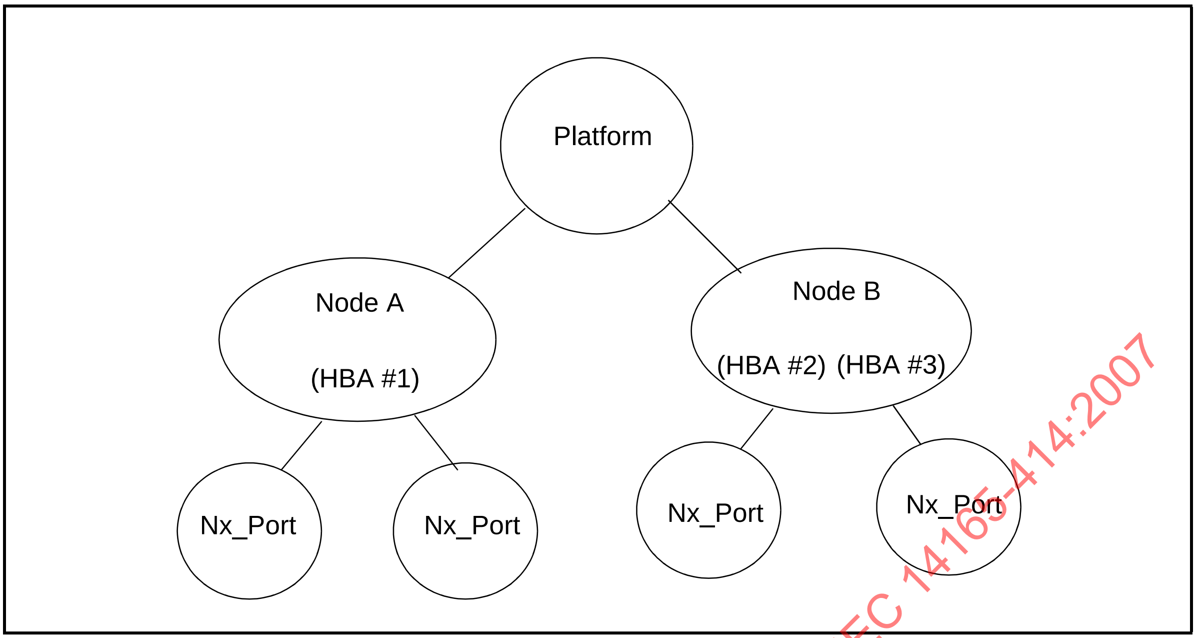


Figure 15 – HBA Management Server Platform Model

6.5.4.3 Protocol

6.5.4.3.1 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.

Synchronous transactions shall be used, as defined in 4.5.2. 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.

The action of the HBA Management Server is unaffected by server sessions.

6.5.4.3.2 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:

- GS_Subtype: as indicated in table 368 and

- Command Code: see table 369 for Request command codes.

Table 369 – HBA Management Server - Request Command Codes

Code (hex)	**Mnemonic**	**Description**	**Attribute(s) in Request CT_IU**	**Attribute(s) in Accept CT_IU**
0100	GRHL	Get Registered HBA List	Null	Registered HBA List
0101	GHAT	Get HBA Attributes	HBA Identifier	Registered Port List, HBA Attribute Block
0102	GRPL	Get Registered Port List	HBA Identifier	Registered Port List
0110	GPAT	Get Port Attributes	Port_Name	Port Attribute Block
0200	RHBA	Register HBA	HBA Identifier, Registered Port List, HBA Attribute Block	none
0201	RHAT	Register HBA Attributes	HBA Identifier, HBA Attribute Block	none
0210	RPRT	Register Port	HBA Identifier, Port_Name, Port Attribute Block	none
0211	RPA	Register Port Attributes	Port_Name, Port Attribute Block	none
0300	DHBA	Deregister HBA	HBA Identifier	none
0301	DHAT	Deregister HBA Attributes	HBA Identifier	none
0310	DPRT	Deregister Port	Port_Name	none
0311	DPA	Deregister Port Attributes	Port_Name	none

6.5.4.3.3 Registration

6.5.4.3.3.1 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, along with an attribute to be associated with the object.

The registration requests defined for the HBA Management Server are summarized in table 369. 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.

6.5.4.3.3.2 Registration rules

The rules for HBA and Port registration shall be as follows:

a) if a registration or deregistration operation does not originate from a Port contained in the associated HBA's Registered Port list, then the request is failed with the appropriate reason code explanation;

NOTE For initial registration of an HBA the registered port list is included in the registration request.

b) if a registration for an HBA or Port is requested and the specified HBA or 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 Ports with the HBA Management Server, the HBA or Port shall be de-registered prior to a subsequent registration taking place;

d) HBA and Port attributes may be modified without de-registering the 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.

6.5.4.3.4 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 369.

6.5.4.4 HBA Management Server Attributes

6.5.4.4.1 Overview

The HBA Management Server includes four types of HBA Management Information:

- Host Bus Adapter Attributes;
- Port attributes;
- Port Statistics and Counters and

- FC-4 Information (e.g. FCP).

Each of these types of HBA Management Information is discussed in this subclause.

6.5.4.4.2 Host Bus Adapter Attributes

6.5.4.4.2.1 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 a “x to y byte printable ASCII String”. Such values are a concatenation of a number *n* of octets, the number *n* being equal to or greater than *x* and equal to or less than *y*, and the value of each octet being equal to or greater than 20h and equal to or less than 7Eh.

NOTE 1 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)

NOTE 2 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.

6.5.4.4.2.2 Manufacturer

The Manufacturer Attribute contains a 1 to 63 byte 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.

6.5.4.4.2.3 Serial Number

The Serial Number Attribute contains a 1 to 63 byte 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.

6.5.4.4.2.4 Model

The Model Attribute contains a 1 to 255 byte 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.

6.5.4.4.2.5 Model Description

The Model Description Attribute contains a 1 to 255 byte 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.

6.5.4.4.2.6 Node_Name

The Node_Name attribute contains an 8 byte value that identifies the Node that contains the ports on the host adapter. The format of the Node_Name attribute shall be the format of the Name_Identifier described in FC-FS. The Node_Name attribute shall be registered with the HBA Management Server as an HBA attribute for each HBA registered with the Fabric. If any ports on the host bus adapter are registered with the Name Server, the Node_Name attribute registered with the HBA Management Server for the host bus adapter shall match the Node_Name attribute registered with the Name Server for its ports. This attribute ties the HBA Management Information defined in the HBA Management Server with the HBA information specified in the Name Server.

6.5.4.4.2.7 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.

6.5.4.4.2.8 Hardware Version

The Hardware Version attribute contains a 1 to 255 byte 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.

6.5.4.4.2.9 Driver Version

The Driver Version attribute contains a 1 to 255 byte 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.

6.5.4.4.2.10 Option ROM Version

The Option ROM Version attribute contains a 1 to 255 byte 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.

6.5.4.4.2.11 Firmware Version

The Firmware Version attribute contains a 1 to 255 byte 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.

6.5.4.4.2.12 Operating System Name and Version

The Operating System Name and Version attribute contains a 1 to 255 byte 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.

6.5.4.4.2.13 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 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.

6.5.4.4.3 Port attributes

6.5.4.4.3.1 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 "*x* to *y* byte Printable ASCII String". Such values are a concatenation of a number *n* of octets, the number *n* being equal to or greater than *x* and equal to or less than *y*, and the value of each octet being equal to or greater than 20h and equal to or less than 7Eh.

NOTE 1 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)

NOTE 2 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.

6.5.4.4.3.2 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.

6.5.4.4.3.3 Fibre Channel Identifier

The Port Fibre Channel 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.

6.5.4.4.3.4 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.

6.5.4.4.3.5 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.

6.5.4.4.3.6 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 FC_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.

6.5.4.4.3.7 Active FC-4 Types

The 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 FC_Port shall register a Name Server FC-4 TYPEs value that indicates support for any FC-4 TYPE that the FC_Port is completely configured to support.

6.5.4.4.3.8 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 370 the Fibre Channel Transmission Speeds that are supported on the specified Nx_Port. The Supported Speed attribute is registered with the HBA Management Service as a Port attribute.

Table 370 – Transmission Speed Mask Values

Mask Value (hex)	**Fibre Channel Transmission Speed**
0000 0001	1 Gbit/s
0000 0002	2 Gbit/s
0000 0004	10 Gbit/s
0000 0008	4 Gbit/s
0000 8000	Speed has not been established or is not able to be determined
other values	Reserved

6.5.4.4.3.9 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 370 the Fibre Channel Transmission Speed at which the specified Nx_Port is currently operating. The value of this attribute shall contain only a single mask value from table 370. The 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 Port Speed attribute shall indicate the speed determined by the HBA Management Server, regardless of any value registered.

6.5.4.4.3.10 Maximum Frame Size

The Maximum Frame Size attribute contains a 32-bit unsigned integer whose value 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.

6.5.4.4.3.11 OS Device Name

The OS Device Name attribute contains a 1 to 255 byte 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.

6.5.4.4.3.12 Host Name

The Host Name attribute contains a 1 to 255 byte Printable ASCII String that describes the name of the host associated with the 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.

6.5.4.4.4 Port Statistics

Port statistics and counters are currently defined by the HBA Management API. Due to their dynamic nature, these attributes are currently not addressed in this standard.

6.5.4.4.5 FC-4 Information

FC-4 related information is defined by the HBA Management API (e.g. FCP). These attributes are currently not addressed in this standard.

6.5.4.5 HBA Management Server Objects

6.5.4.5.1 HBA Identifier

An HBA Identifier shall consist of a Port_Name associated with the specific HBA. When an HBA has only one associated Port, then the HBA Identifier shall be the Port_Name of that single Port. For HBAs with multiple associated Ports, the HBA Identifier shall be one of the HBA’s associated Port_Names. Once an HBA has chosen a Port_Name for identification, the chosen Port_Name shall be referred to as the HBA’s Identification Port_Name and that Port_Name shall consistently (e.g., across power cycles) be used to uniquely identify the HBA.

The format of the HBA Identifier is depicted in table 371 below.

Table 371 – HBA Identifier

Item	**Size** bytes
Identification Port_Name	8

6.5.4.5.2 Registered Port List

The format of the Registered Port List is shown in table 372 below.

Table 372 – 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 373 below.

Table 373 – Port Entry

Item	**Size** bytes
Port_Name	8

6.5.4.5.3 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 374 below.

Table 374 – 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* to 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 6.5.4.5.4.

6.5.4.5.4 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 375.

Table 375 – Attribute Entry

Item	**Size** bytes
Attribute Entry Type	2
Attribute Entry Length	2
Attribute Entry Value	see table 376

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 376 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 376.

Table 376 – Attribute Entry Types and associated Values

Type (hex)	**Value**			
	Description	**Length**	**Type**	**Required**
0001	Node_Name	8 bytes	Binary	Yes
0002	Manufacturer	4 to 64 bytes	ASCII	No
0003	Serial Number	4 to 64 bytes	ASCII	No
0004	Model	4 to 256 bytes	ASCII	No
0005	Model Description	4 to 256 bytes	ASCII	No
0006	Hardware Version	4 to 256 bytes	ASCII	No
0007	Driver Version	4 to 256 bytes	ASCII	No
0008	Option ROM Version	4 to 256 bytes	ASCII	No
0009	Firmware Version	4 to 256 bytes	ASCII	No
000A	OS Name and Version	4 to 256 bytes	ASCII	No
000B	Maximum CT Payload Length	4 bytes	Binary	No
other values	Reserved			

6.5.4.5.5 Port Attribute Block

The Port Attribute Block is a variable length structure that contains attributes registered for the specified Port. The format of the Port Attribute Block is depicted in the table 377 below.

Table 377 – 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* to 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 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 6.5.4.5.4.

6.5.4.5.6 Port Attributes

The valid Port Attributes are depicted in table 378.

Table 378 – Port Attribute Values

Type (hex)	**Description**	**Length**	**Type**	**Required**
0001	Supported FC-4 Types	32 bytes	Bitmask	No
0002	Supported Speed	4 bytes	Bitmask	No
0003	Current Port Speed	4 bytes	Bitmask	No
0004	Maximum Frame Size	4 bytes	Binary	No
0005	OS Device Name	(1 to 256) bytes	ASCII	No
0006	Host Name	(1 to 256) bytes	ASCII	No
other values	Reserved			

6.5.4.6 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 379, is returned.

Table 379 – Reason Code Explanations

Encoded value (hex)	**Description**
00	No Additional Explanation
10	HBA Already Registered
11	Attributes For Specified HBA Not Registered
12	HBA Attribute Block Contains Multiple Attributes of the Same Type
13	Invalid HBA Attribute Block Length
14	Required HBA Attributes Not Present
15	Originating Port Not in Registered Port List
16	HBA Identifier Not in Registered Port List
17	HBA Not Registered
20	Port Attributes Not Registered
21	Port Not Registered
22	Port Attribute Block Contains Multiple Attributes of the Same Type
23	Invalid Port Attribute Block Length
24	Port Already Registered
others	Reserved

6.5.4.7 Commands

6.5.4.7.1 Query – Get Registered HBA List (GRHL)

The HBA Management Server shall, when 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 380 below.

Table 380 – GRHL Request Payload

Item	**Size** bytes
CT_IU preamble	see 4.3

The format of the GRHL Accept payload is depicted in table 381 below.

Table 381 – 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

6.5.4.7.2 Query – Get HBA Attributes (GHAT)

The HBA Management Server shall, when 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 382 below.

Table 382 – 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 383 below.

Table 383 – GHAT Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3
Registered Port List	see table 372
HBA Attribute Block	see table 374

6.5.4.7.3 Query – Get Registered Port List (GRPL)

The HBA Management Server shall, when it receives a GRPL request, return a list of registered Ports for the specified HBA. The GRPL request payload shall specify the HBA Identifier for which the registered ports are sought. The GRPL accept payload shall contain the Registered Port list.

The format of the GRPL Request payload is depicted in table 384 below.

Table 384 – 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 385 below.

Table 385 – GRPL Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3
Registered Port List	see table 372

Registered Port List: The Registered Port list is described in 6.5.4.5.2.

6.5.4.7.4 Query – Get Port Attributes (GPAT)

The HBA Management Server shall, when it receives a GPAT request, return the Port Attribute Block for the specified Port. The GPAT request payload shall specify the name identifying the Port for which attributes are sought. The GPAT accept payload contains the Attribute Block for the Port specified in the request.

If the specified 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 386 below.

Table 386 – 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 387 below.

Table 387 – GPAT Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3
Port Attribute Block	see table 377

Port_Name: This field specifies the name of the Port specified in the GPAT Request.

Port Attribute Block: The format of the Port Attribute Block is described in 6.5.4.5.5.

6.5.4.7.5 Register HBA (RHBA)

The HBA Management Server shall when it receives an RHBA request, register the HBA, its 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 Registered Port List shall contain the Port_Name that is used as the HBA Identifier. If the Registered Port list does not contain the Port_Name used as the HBA Identifier then the request is failed with the appropriate reason code explanation.

The Node_Name attribute shall be required in the HBA Attribute Block. If the Node_Name attribute is not specified in the HBA Attribute Block then the request is failed with the appropriate reason code explanation.

The RHBA request shall originate from a 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 a 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 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 388 below.

Table 388 – RHBA Request Payload

Item	**Size** bytes
CT_IU preamble	see 4.3
HBA Identifier	8
Registered Port List	see table 372
HBA Attribute Block	see table 374

HBA Identifier: The format of the HBA Identifier is described in 6.5.4.5.1.

Registered Port List: The format of the Registered Port list is described in 6.5.4.5.2.

HBA Attribute Block: The format of the HBA Attribute block is described in 6.5.4.5.3.

The format of the RHBA Accept payload is depicted in table 389 below.

Table 389 – RHBA Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3

6.5.4.7.6 Register HBA Attributes (RHAT)

The HBA Management Server shall, when 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 Node_Name attribute shall be required in the HBA Attribute Block. If the Node_Name attribute is not specified in the HBA Attribute Block then the request is failed with the appropriate reason code explanation.

The format of the RHAT Request payload is depicted in table 390 below.

Table 390 – RHAT Request Payload

Item	**Size** bytes
CT_IU preamble	see 4.3
HBA Identifier	8
HBA Attribute Block	see table 374

HBA Identifier: The format of the HBA Identifier is described in 6.5.4.5.1.

HBA Attribute Block: The format of the HBA Attribute block is described in 6.5.4.5.3.

The format of the RHAT Accept payload is depicted in table 391 below.

Table 391 – RHAT Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3

6.5.4.7.7 Register Port (RPRT)

The HBA Management Server shall, when it receives an RPRT request, register the specified Port and its attributes contained in the Port Attribute Block for the specified Port. In addition, the 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 specified attributes are unable to be registered for the Port, then the 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 392 below.

Table 392 – RPRT Request Payload

Item	**Size** bytes
CT_IU preamble	see 4.3
HBA Identifier	8
Port_Name	8
Port Attribute Block	see table 377

HBA Identifier: The format of the HBA Identifier is described in 6.5.4.5.1.

Port_Name: The name of the Port for which attributes are to be registered.

Port Attribute Block: The format of the Port Attribute block is described in 6.5.4.5.5.

The format of the RPRT accept payload is depicted in table 393 below.

Table 393 – RPRT Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3

6.5.4.7.8 Register Port Attributes (RPA)

The HBA Management Server shall, when it receives an RPA request, register the attributes contained in the Port Attribute block for the specified Port. If the specified 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 394 below.

Table 394 – RPA Request Payload

Item	**Size** bytes
CT_IU preamble	see 4.3
Port_Name	8
Port Attribute Block	see table 377

Port_Name: The name of the Port for which attributes are to be registered.

Port Attribute Block: The format of the Port Attribute block is described in 6.5.4.5.5.

The format of the RPA accept payload is depicted in table 395 below.

Table 395 – RPA Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3

6.5.4.7.9 Deregister HBA (DHBA)

The HBA Management Server shall, when it receives an DHBA request, de-register the HBA and its attributes. In addition, all 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 396 below.

Table 396 – 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 397 below.

Table 397 – DHBA Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3

6.5.4.7.10 Deregister HBA Attributes (DHAT)

The HBA Management Server shall, when 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 398 below.

Table 398 – 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 399 below.

Table 399 – DHAT Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3

6.5.4.7.11 Deregister Port (DPRT)

The HBA Management Server shall, when it receives an DPRT request, de-register the specified 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 Port to be de-registered. The DPRT accept payload shall be null.

If the specified Port is the last registered port for the HBA, then the HBA and its attributes are also de-registered.

If the specified 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 400 below.

Table 400 – 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 401 below.

Table 401 – DPRT Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3

6.5.4.7.12 Deregister Port Attributes (DPA)

The HBA Management Server shall, when it receives a DPA request, de-register all optional attributes for the specified Port. The DPA request payload shall specify the Port_Name that identifies the Port for which its attributes are to be de-registered.

If attributes for the Port have not been registered with the HBA Management Server, then the request is rejected with the appropriate reason code explanations.

If the DPA does not originate from a Port contained in the HBA's Registered Port list, then the request is failed with the appropriate reason code explanation.

The format of the DPA Request payload is depicted in table 402 below.

Table 402 – 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 403 below.

Table 403 – DPA Accept Payload

Item	**Size** bytes
CT_IU preamble	see 4.3

7 Alias Service

7.1 Overview

The Alias Service manages the registration and deregistration of Alias IDs for both Hunt Groups and Multicast Groups. The Alias Service is not involved in the routing of frames for any Group.

The Alias Service may be internal or external to the Fabric, but, in either case, it is addressed by means of the WKA identifier, FFFFF8h. This clause describes the registration/deregistration process in more detail.

Authorization for Alias Service operations is provided.

The GS_Type for all Alias Services shall be set as indicated in table 4. Table 404 defines the GS_Subtype codes for the Alias Service.

Table 404 – Alias Service subtype values

Values (hex)	**Description**
01	Alias Server
other values	Reserved

7.2 Alias Server

7.2.1 Overview

Alias registration and deregistration are managed through protocols containing a set of request/reply IUs supported by the Alias Server.

7.2.2 Protocol

7.2.2.1 Overview

Synchronous transactions shall be used. For an Alias Server request, the Alias Server payload shall be transported from the requestor to the Alias Server using a Request CT_IU. The corresponding Alias Server response is transported from the Alias Server to the requestor, in the Exchange established by the requestor, using a Response CT_IU.

The action of the Alias 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 Name Server request and their responses. Fields not specified here shall be set as defined in 4.3.2.

- GS_Subtype: This field shall be set to 01h and

- Command Code: see table 405 for Request command codes.

Table 405 – Alias Server Requests

Code hex	**Mnem.**	**Description**
0001	JNA	Join Alias Group
0002	RMA	Remove from Alias Group
0003	LSN	Listen
0004	SLSN	Stop Listen
0005	RAG	Read Alias Group

7.2.2.3 Alias server functions

7.2.2.3.1 Overview

The following subclauses describe the functions performed by the Alias Server for each of the supported requests. Figure 16 illustrates the flow among the Originator Nx_Port, participating Nx_Ports, Alias Server and Fabric Controller to create an Alias Group.

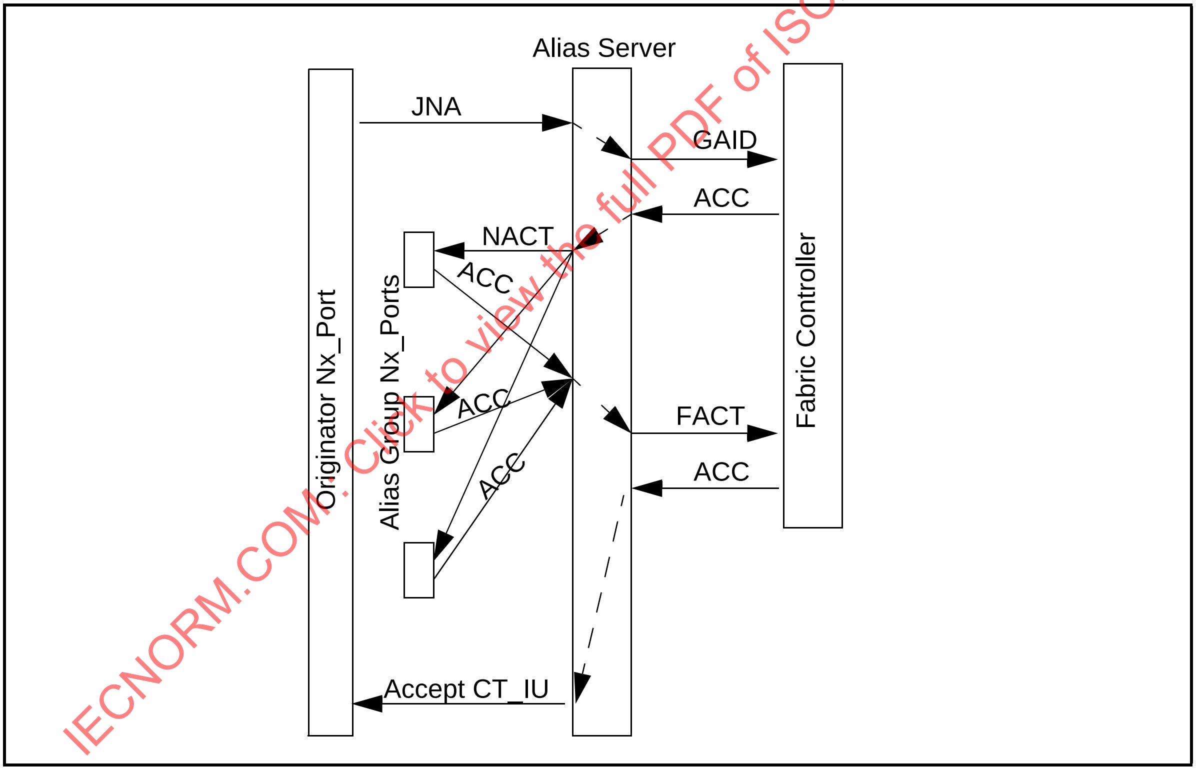


Figure 16 –Function flow

7.2.2.3.2 Join alias group

Upon receipt of a Join Alias Group request, the Alias Server shall perform the following functions, in the specified order.

a) The Alias Server shall reject the request with a Reason Code of “Unable to perform command request” and a Reason Code Explanation of “Invalid Alias_Token” if the passed Alias_Token is not valid.

b) The Alias Server shall reject the request with a Reason Code of “Unable to perform command request” and a Reason Code Explanation of “Unsupported Alias_Token” if the Flags in the passed Alias_Token indicate an Alias Group that is not supported.

c) The Alias Server shall determine whether or not the specified Alias Group has already been created.

d) If the Alias Group has not already been created, an attempt is made to create a new Alias Group.

   The request shall be rejected with a Reason Code of “Unable to perform command request” and a Reason Code Explanation of “Invalid Authorization_Control” if the passed Authorization_Control is not valid;

   If a Multicast Group is being formed and the Alias_SP do not contain valid Class 3 Service Parameters, the request shall be rejected with a Reason Code of “Unable to perform command request” and a Reason Code Explanation of “Alias Group cannot be formed (Invalid Class)”.

   The Alias Server shall obtain a unique alias address identifier for this Alias Group from the Fabric Controller either with a Fabric Controller Request, Get Alias Group ID (GAID) or an implicit mechanism. The Alias_Token is passed in the payload of the request and the reply returns the assigned alias address identifier. If the Fabric returns an LS_RJT, the Join Alias Group is rejected with the same Reason Code Explanation as was contained in the LS_RJT to the GAID.

   The passed Authorization_PW and Authorization_Control are attached to the defined Alias Group.

e) If the Alias Group has already been created, an attempt is made to modify the Alias Group.

   The request shall be rejected with a Reason Code of “Unable to perform command request” and a Reason Code Explanation of “Unauthorized Request (Invalid Password)” if the Authorization_PW of the indicated Alias Group is non-zero and does not match the passed Authorization_PW.

   The request shall be rejected with a Reason Code of “Unable to perform command request” and a Reason Code Explanation of “Unauthorized Request (Invalid Authorization_Control)” if the Authorization_Control attached to the passed Alias Group does not permit the initiating Nx_Port to add the Nx_Ports in the passed NP_List.

f) The Alias Server may send an Extended Link Services request, Nx_Port Activate Alias Group ID (NACT), to each of the Nx_Ports in the passed list (see FC-FS).

   If the Alias Group is being created, and other Nx_Ports are allowed to Listen, then the Alias Server may also send a NACT to all Nx_Ports that have registered to listen to the Alias Class matching the Alias Class of the Alias Group being created (if any).