5 Directory Service

5.1 Overview

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

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

Table 19 defines the GS_Subtype codes for the Directory Service.

Table 19 – Directory Service subtype values

Values (hex)	Description
01	Obsolete[a]
02	Name Server
03	Obsolete[b]
80-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 (i.e., Zone) of the node.

5.2 Name Server

5.2.1 Overview

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 Third party registration support is obsolete.

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

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

Table 20 – Name Server - request types

Command Code (hex)	Description
01xx	Get object(s) (Query)
02xx	Register object
03xx	Deregister object(s)
04xx	delimiter based action object(s)

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

Table 21 – Name Server - objects

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

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

5.2.2 Name Server protocol

5.2.2.1 Overview

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

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

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

5.2.2.2 CT_IU preamble values

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

a) GS_Subtype: as indicated in table 19; and

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

Table 22 – Name Server - request Command Codes

Code (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	obsolete		

Table 22 – 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	obsolete		
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	obsolete		
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	obsolete		
0136	GIPA_NN	obsolete		
0139	GSNN_NN	Get Symbolic Node Name	Node Name	Symbolic Node Name
0153	GNN_IP	obsolete		
0156	GIPA_IP	obsolete		
0171	GID_FT	Get Port Identifiers - FC-4 type	Flags[a]	List of Port Identifiers
0172	GPN_FT	Get Port Names - FC-4 type	Flags[a]	List of Port Identifiers and Port Names
0173	GNN_FT	Get Node Names - FC-4 type	Flags[a]	List of Port Identifiers and Node Names
0180	GNN_FF	Get Node Names - FC-4 Features	FC-4 Features, Flags	List of Port Identifiers and Node Names
0181	GPN_FF	Get Port Names - FC-4 Features	FC-4 Features, Flags	List of Port Identifiers and Port Names
0182	GPN_SDFCP	Get Port Names - Simplified Discovery for FCP	FC-4 Features	List of Port Identifiers and Port Names

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

Code (hex)	mnemonic	Description	object(s) in request CT_IU	object(s) in Accept CT_IU
01A1	GID_PT	Get Port Identifiers - Port Type	Port Type, Flags	List of Port Identifiers
01B1	GID_IPP	obsolete		
01B2	GPN_IPP	obsolete		
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, Flags	List of Port Identifiers
01F2	GID_DP	Get Port Identifier - Domain/Port	Domain/Port	Port Identifier
0212	RPN_ID	obsolete		
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	obsolete		
021B	RIPP_ID	obsolete		
021D	RHA_ID	Register Hard Address - Port Identifier	Port Identifier, Hard Address	none
021E	RFD_ID	obsolete		
021F	RFF_ID	Register FC-4 Features - Port Identifier	Port Identifier, FC-4 Features	none
0235	RIP_NN	obsolete		
0236	RIPA_NN	obsolete		
0239	RSNN_NN	Register Symbolic Node Name	Node Name, Symbolic Node Name	none

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

Code (hex)	mnemonic	Description	object(s) in request CT_IU	object(s) in Accept CT_IU
0300	DA_ID	De-register all	Port Identifier	none
0400	DBA	delimiter based action[d]	delimiter objects	delimiter objects
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 is presented in 4.9. Interaction of these requests with the name server is documented in 5.2.2.7.

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

5.2.2.3 Registration

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

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

Registration requests shall be processed:

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

b) from the Fabric Controller; and

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

The Name Server shall reject registrations from any other source.

The Name Server may reject registrations:

a) due to Name Server resource limitations; or

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

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

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

a) the address identifier 00 00 00h;

b) WKA identifiers; or

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

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

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

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

a) Port Type;

b) Port Identifier;

c) Port Name;

d) Node Name;

e) Class of Service; or

f) Permanent Port Name.

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

5.2.2.4 Queries

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

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

5.2.2.5 Deregistration

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

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

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

5.2.2.6 Delimiter based actions

5.2.2.6.1 Overview

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

5.2.2.6.2 Delimiter based action request

Each delimiter based action request shall contain:

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

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

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

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

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

2) one or more key data TLVs; and

3) one or more non-key data TLVs.

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

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

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

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

2) one or more key data TLVs;

3) one query parameters TLV;

4) one or more non-key data TLVs.

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

5.2.2.6.3 Delimiter based action response

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

The delimiter based action response shall include:

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

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

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

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

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

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

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

   and

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

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

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

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

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

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

3) one query parameters TLV; and

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

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

5.2.2.7 Name Server Interaction with Common requests SSB and SSE

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

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

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

5.2.3 Name Server objects - formats

5.2.3.1 Use of Null values for Name Server objects

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

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

5.2.3.2 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-FS-3.

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-FS-3.

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-FS-3.

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

5.2.3.5 Class of Service - format

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

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

Bit 1 - obsolete

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

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

Bit 4 - obsolete

Bit 5: reserved

Bit 6 - obsolete

Bits 7 to 31: reserved

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

5.2.3.6 FC-4 types - format

The FC-4 types object indicates a set of supported 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-3). 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 23 – FC-4 types mapping

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

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

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

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

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

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

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

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

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

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

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

and

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

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

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

5.2.3.7 Symbolic Port Name - format

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

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

5.2.3.8 Symbolic Node Name - format

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

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

5.2.3.9 Port Type - format

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

Table 24 – Port Type - encoding

Encoded value (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.

5.2.3.10 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-3. The Fabric Port Name for a given Port Identifier is the Port_Name for the Fx_Port to which the PN_Port through which the Nx_Port communicates is attached.

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

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

5.2.3.11 Hard Address - format

The format of the Hard Address object, 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-3).

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

5.2.3.12 FC-4 Features - format

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

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

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

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

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

   D) 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 25); and

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

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

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

Table 25 – FC-4 Features mapping

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

5.2.3.13 Permanent Port Name - format

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

The format of the Permanent Port Name object, as used by the Name Server, shall be identical to the Name_Identifier format defined in FC-FS-3.

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

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

5.2.3.14 Delimiter based action - format

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

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

Table 26 – Delimiter based action - format

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

5.2.3.15 Delimiter based action header - format

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

The format of the delimiter flags field is shown in table 27.1

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

Table 27 – Delimiter based action header - format

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

Table 27.1 - Delimiter flags - format

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

5.2.3.16 Delimiter set - format

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

The format of the delimiter set used when an action type of register or deregister is speicified is shown in table 28.

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

1) one delimiter TLV;

2) one or more key data TLVs; and

3) one or more non-key data TLVs.

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

Table 28 – Delimiter set - register and deregister format

Item	Size (Bytes)
delimiter TLV	8
key data TLV 1	see 5.2.3.18
key data TLV m	see 5.2.3.18
non-key data TLV 1	see 5.2.3.18
non-key data TLV n	see 5.2.3.18

The format of the delimiter set used when an action Type of query is speicified is shown in table 29.

Each delimiter set used to perform query shall contain:

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

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

Table 29 – delimiter set - query format

Item	Size (Bytes)
delimiter TLV	8
key data TLV 1	see 5.2.3.18
key data TLV m	see 5.2.3.18
query parameters TLV	6
non-key data TLV 1	see 5.2.3.18
non-key data TLV n	see 5.2.3.18

5.2.3.17 delimiter TLV - format

The format of the delimiter TLV is shown in Table 30.

The type field shall be set 60h

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

Table 30 – delimiter TLV format

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

The format of the action flags field is shown in Table 30.1.

Table 30.1 - Action flags - format

Bit Position	Description
31-24	Action code. This field indicates the action being requested within the delimiter set: - 0000 0001 = Query - 0000 0010 = Register - 0000 0011 = De-Register - 0000 0100 = Vendor specific
23	Key logical relationship flags - Logical AND. This bit shall be set to zero when an action type of register or De-Register is specified. This bit may be set to one when an action type of query or vendor specific is specified. When an action type of query is specified and this bit is set to one and the logical NOT bit is set to zero, the device performing the query is requesting that a logical AND be performed against all key data TLVs and only those entities registered with the Name Server that match all key data TLV(s) be returned. When an action type of query is specified and this bit is set to one and the logical NOT bit is set to one, the device performing the Query is requesting that a logical NAND be performed against all key data TLVs and only those entities registered with the Name Server that do not match all key data TLV(s) be returned. When an action type of Vendor Specific is specified, the meaning of this bit is undefined. The logical AND and logical OR bits are mutually exclusive and shall not be set to one at the same time. If both the AND and OR bits are set to one, the Name Server shall reject the delimiter set by returning a single data TLV status response TLV with a Reason Code / Reason code explanation of logical error (03)/ No additional explanation (00).

Table 30.1 - Action flags - format

Bit Position	Description
22	Key logical relationship flags - Logical OR. This bit shall be set to zero when an action type of register or de-register is specified. This bit may be set to one when an action type of query or vendor specific is specified. When an action type of query is specified and this bit is set to one and the logical NOT bit is set to zero, the device performing the query is requesting that a logical OR be performed against all key data TLVs and those entities registered with the Name Server that match any key data TLV be returned. When an action type of query is specified and this bit is set to one and the logical NOT bit is set to one, the device performing the query is requesting that a logical NOR be performed against all key data TLVs and only those entities registered with the Name Server that do not match any key data TLV be returned. When an action type of vendor specific is specified, the meaning of this bit is undefined. The logical AND and logical OR bits are mutually exclusive and shall not be set to one at the same time. If both the AND and OR bits are set to one, the Name Server shall reject the delimiter set by returning a single data TLV status response TLV with a Reason Code / Reason code explanation of logical error (03)/ No additional explanation (00).
21	Key logical relationship flags - Logical NOT. This bit shall be set to zero when an action type of register or de-register is specified. This bit may be set to one when an action type of query or vendor specific is specified and either the logical AND or logical OR bits are set. The meaning of this bit is defined in the logical AND and logical OR definitions.
20-16	Key logical relationship flags - Reserved
15	Action Flags - Error All. When this bit is set to one, the device is requesting that if any of the key data or non-key data TLVs contain information that is either unrecognized or cannot be processed for any reason; the command defined in the delimiter set should fail completely and no information will be registered, deregistered or returned from the name server other than error information. When this bit is set to zero, the device is requesting that if any of the key data or non-key data TLVs contain information that is either unrecognized or cannot be processed for any reason; the command defined in the delimiter set should complete on a best effort basis. The key data or non-key data TLVs that contained unrecognizable information shall be indicated by an error code in the command response.
14-0	Action Flags - Reserved

5.2.3.18 data TLV - format

The format of the data TLV is shown in figure 31.

Table 31 – data TLV - format

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

The format of the type field is shown in figure 31.1

Table 31.1 - type Field - format

Bit Position	Description
15	Key. The key bit may be set to one if the object mnemonic type is allowed to be a Registration key (see table 31). When the key bit is set to one, the data TLV is referred to as a key data TLV. When the key bit is set to zero, the data TLV is referred to as a non-key data TLV. Key data TLVs must be listed first in a delimiter set followed by non-Key data TLVs if present. If a key data TLV is listed after a non-key data TLV, the Name Server shall reject the delimiter set by returning a single data TLV status response TLV with a Reason Code / Reason code explanation of logical Error (03)/ No additional explanation (00).
14 - 0	object mnemonic type code (see table 32)

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

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

5.2.3.19 Object mnemonic type code - format

The type field of data TLVs used in delimiter based action commands shall be set to one of the object mnemonic type codes defined in table 32. Only those object mnemonic type codes containing a Y in the “Reg Key Allowed” field may be used when a dataTLVs key bit is set to one. Object mnemonic type codes that do not contain a Y in the “Reg Key Allowed” field shall not be used when a dataTLVs key bit is set to one. If the data TLV key bit is set to one and the object mnemonic type code does not contain a Y in the “Reg Key Allowed” field, the Name Server shall reject the delimiter set by returning a single data TLV status response TLV with a Reason Code / Reason code explanation of logical Error (03)/ No additional explanation (00).

Table 32 – object mnemonic Type Codes

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

5.2.3.20 Query parameters TLV - format

The format of the query parameters TLV is shown in table 33.

Table 33 – Query parameters TLV - format

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

The format of the type field is shown in table 33.1

Table 33.1 - type field - format

Bit Position	Description
15	Reserved
14 - 0	Object mnemonic type Code = 61h (see table 32)

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

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

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

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

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

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

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

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

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

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

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

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

5.2.3.21 Data TLV status response - format

The format of the data TLV status response is shown in table 34.

Table 34 – Data TLV status response - format

Item	Size (Bytes)
Type	2
Length (words)	2
Reserved	2
Reason code	1
Reason code explanation	1

The format of the type field is shown in table 34.1

Table 34.1 - Data TLV status response type field - format

Bit Position	Description
15	Reserved
14-0	Object mnemonic type Code = 62h (see table 32)

The length field shall be set to 02h.

The Reason Code field shall be set to an appropriate Reason Code (see Table 10).

The Reason code explanation field shall be set to an appropriate Reason code explanation (see Table 35)

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 35 is returned.

Table 35 – 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	Obsolete
06	Obsolete
07	FC-4 types not registered
08	Symbolic Port Name not registered
09	Symbolic Node Name not registered
0A	Port Type not registered
0B	Obsolete

Table 35 – Reject CT_IU Reason code explanations

Encoded value (hex)	Description
0C	Fabric Port Name not registered
0D	Hard Address not registered
0E	Obsolete
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 the specified object is not registered;

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

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

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

5.2.5 Commands

5.2.5.1 Overview

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

5.2.5.2 Query - Get all next (GA_NXT)

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

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

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

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

The format of the GA_NXT request CT_IU is shown in table 36. 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 36 – 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 37. The format of the various objects returned is defined in 5.2.3.

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

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

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

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

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

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

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

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

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

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

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

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

Table 37 – Accept CT_IU to GA_NXT request

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

5.2.5.3 Query - Get identifiers (GID_A)

The Name Server shall, 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 38. The requestor supplies a Domain_ID Scope that defines the scope for which identifiers are sought.

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

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

Table 38 – 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 39 and table 40.

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

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

b) Bits 6-0 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 39 – 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 40 – 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 r)	1
requested Domain_ID #1	1
Area_ID #1	1
reserved	1
...	
Control (1 r 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 41. The requestor supplies the Port Identifier for which the Port Name is sought.

Table 41 – 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 42.

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

Table 42 – 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 43. The requestor supplies the Port Identifier for which the Node Name is sought.

Table 43 – 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 44.

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

Table 44 – 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 45. The requestor supplies the Port Identifier for which the Class of Service object is sought.

Table 45 – 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 46.

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

Table 46 – 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 47. The requestor supplies the Port Identifier for which the FC-4 types object is sought.

Table 47 – 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 48.

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

Table 48 – 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 49. The requestor supplies the Port Identifier for which the Symbolic Port Name is sought.

Table 49 – 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 50.

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 50 – Accept CT_IU to GSPN_ID request

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

5.2.5.9 Query - Get Port Type (GPT_ID)

The Name Server shall, 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 51. The requestor supplies the Port Identifier for which the Port Type is sought.

Table 51 – 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 52.

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

Table 52 – Accept CT_IU to GPT_ID request

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

5.2.5.10 Query - Get Fabric Port Name (GFPN_ID)

The Name Server shall, 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 53. The requestor supplies the Port Identifier for which the Fabric Port Name is sought.

Table 53 – 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 54.

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

Table 54 – Accept CT_IU to GFPN_ID request

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

5.2.5.11 Query - Get Hard Address (GHA_ID)

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

Table 55 – 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 56.

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

Table 56 – Accept CT_IU to GHA_ID request

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

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

The GFD_ID CT_IU request is obsolete.

5.2.5.13 Query - Get FC-4 Features (GFF_ID)

The Name Server shall, 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 57. The requestor supplies the Port Identifier for which the FC-4 Features object is sought.

Table 57 – 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 58.

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

Table 58 – Accept CT_IU to GFF_ID request

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

5.2.5.14 Query - Get Port Identifier (GID_PN)

The Name Server shall, 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 59. The requestor supplies the Port Name for which the Port Identifier is sought.

Table 59 – 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 60.

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

Table 60 – Accept CT_IU to GID_PN request

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

5.2.5.15 Query - Get Port Identifiers (GID_NN)

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

Table 61 – 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 62.

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 6-0 are reserved.

Table 62 – Accept CT_IU to GID_NN request

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

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

5.2.5.16 Query - Get Port Names (GPN_NN)

The Name Server shall, 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 63. The requestor supplies the Node Name for which associated Port Identifiers and Port Names are sought.

Table 63 – 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 64.

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

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

b) Bits 6-0 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 64 – Accept CT_IU to GPN_NN request

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

5.2.5.17 Query - Get Symbolic Node Name (GSNN_NN)

The Name Server shall, 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 65. The requestor supplies the Node Name for which the Symbolic Node Name is sought.

Table 65 – 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 66.

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 66 – Accept CT_IU to GSNN_NN request

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

5.2.5.18 Query - Get Port Identifiers (GID_FT)

The Name Server shall, 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 67. The requestor supplies the FC-4 type code (see FC-FS-3) for which supporting Port Identifiers are sought.The settings of bits in the Flags field affects operation of the request. The format of the Flags field is as follows:

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

b) Bits 6 through 0 - reserved.

NOTE 17 – The type is specified as an 8-bit encoded FC-FS-3 value, not as an FC-4 type object.

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

the Area_ID flag is set to one, the Name Server shall return Port Identifiers within the specified Domain and Area having registered support for the specified FC-4 type code.

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

Table 67 – GID_FT request CT_IU

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

The format of the Accept CT_IU to a GID_FT request is shown in table 68.

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

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

b) Bits 6-0 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 68 – Accept CT_IU to GID_FT request

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

5.2.5.19 Query - Get Port Names (GPN_FT)

The Name Server shall, 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 69. The requestor supplies the FC-4 type code (see FC-FS-3) for which supporting Port Identifiers and Port Names are sought.The settings of bits in the Flags field affects operation of the request. The format of the Flags field is as follows:

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

b) Bits 6 through 0 - reserved.

NOTE 19 – The type is specified as an 8-bit encoded FC-FS-3 value, not as an FC-4 type object.

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

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

Table 69 – GPN_FT request CT_IU

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

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

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

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

b) Bits 6-0 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 70 – Accept CT_IU to GPN_FT request

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

5.2.5.20 Query - Get Node Names (GNN_FT)

The Name Server shall, 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 71. The requestor supplies the FC-4 type code (see FC-FS-3) for which supporting Port Identifiers and Node Names are sought. The settings of bits in the Flags field affects operation of the request. The format of the Flags field is as follows:

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

b) Bits 6 through 0 - reserved.

NOTE 21 – The type is specified as an 8-bit encoded FC-FS-3 value, not as an FC-4 type object.

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

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

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

Table 71 – GNN_FT request CT_IU

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

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

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

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

b) Bits 6-0 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 72 – 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 r)	1
Port Identifier #1	3
reserved	4
Node Name #1	8
...	
Control (1 r r r r r r r r)	1
Port Identifier #n	3
reserved	4
Node Name #n	8

5.2.5.21 Query - Get Node Names (GNN_FF)

The Name Server shall, when it receives a GNN_FF request, return all Port Identifiers and Node Names having registered support for the specified type code value and corresponding FC-4 Feature bits. The format of the GNN_FF request CT_IU is shown in table 73. The requestor supplies the type code and FC-4 Feature bits for which supporting Port Identifiers and Node Names are sought. The format of the FC-4 Feature bits field is:

a) Bits 7-4 are reserved; and

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

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

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

b) Bits 6 through 0 - reserved.

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

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

Table 73 – GNN_FF request CT_IU

Item	Size (Bytes)
CT_UI preamble	see 4.3
Flags	1
Domain_ID scope	1
Area_ID scope	1
reserved	1
reserved	2
FC-4 Feature bits	1
type code	1

The format of the Accept CT_IU to a GPN_FT request is shown in table 74.

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

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

b) Bits 6-0 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 74 – GNN_FF Accept CT_IU (Same as GNN_FT)

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

5.2.5.22 Query - Get Port Names (GPN_FF)

The Name Server shall, when it receives a GPN_FF request, return all Port Identifiers and Port Names having registered support for the specified type code value and corresponding FC-4 Feature bits. The format of the GPN_FF request CT_IU is shown in table 75. The requestor supplies the type code and FC-4 Feature bits for which supporting Port Identifiers and Port Names are sought. The format of the FC-4 Feature bits field is:

a) Bits 7-4 are reserved; and

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

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

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

b) Bits 6 through 0 - reserved.

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

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

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

Table 75 – GPN_FF request CT_IU

Item	Size (Bytes)
CT_UI preamble	see 4.3
Flags	1
Domain_ID scope	1
Area_ID scope	1
reserved	1
reserved	2
FC-4 Feature bits	1
type code	1

The format of the Accept CT_IU to a GPN_FF request is shown in table 76.

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

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

b) Bits 6-0 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 76 – GPN_FF Accept CT_IU

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

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

The Name Server shall, when it receives a GPN_SDFCP request, return Port Identifiers and Port Names for all Nx_Ports having registered support for the FCP type code (i.e., 08h) and FC-4 Feature bits that match the combination specified by the FCP Feature selection bits (see table 79). The format of the GPN_SDFCP request CT_IU is shown in table 77.

Table 77 – GPN_SDFCP request CT_IU

Item	Size (Bytes)
CT_UI preamble	see 4.3
Flags	1
Domain_ID scope	1
Area_ID scope	1
reserved	1
reserved	3
FC-4 Feature selection bits	1

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

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

b) Bits 6 through 0 - reserved.

As shown in table 78, the Domain_ID Scope and Area_ID Scope fields specify the scope of the request.

Table 78 – Domain and Area scope

Domain ID Scope	Area ID scope	Flags (bit 7)	Port Identifiers returned
0	0	0	All Port Identifiers and Port Names having registered the FCP type (i.e., 08h) and the specified FC-4 Feature bits, see table 79.
Non-zero	0	0	Port Identifiers and Port Names within the specified Domain having registered the FCP type code (i.e., 08h) and FC-4 Feature bits, see table 79.
Non-zero	0	1	Port Identifiers and Port Names within the specified Domain and an Area of 00h having registered the FCP type code (i.e., 08h) and FC-4 Feature bits, see table 79.
Non-zero	Non-zero	N/A	Port Identifiers and Port Names within the specified Domain and Area having registered the FCP type code (i.e., 08h) and FC-4 Feature bits, see table 79.

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

The FCP Feature selections bits field shall contain the feature(s) being requested as follows:

c) initiator function selection (bit 1); and

d) target function selection (bit 0).

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

The initiator function bit and target function bit in table 79 shall be as defined in FCP-4.

The FC-4 Feature selection field usage is shown in table 79.

Table 79 – Feature selection bit usage

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

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

Table 80 – Accept CT_IU to GPN_SDFCP request

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

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

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

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

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

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

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

5.2.5.24 Query - Get Port Identifiers (GID_PT)

The Name Server shall, 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 81. The requestor supplies the Port Type for which supporting Port Identifiers are sought. The settings of bits in the Flags field affects operation of the request. The format of the Flags field is as follows:

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

b) Bits 6 through 0 - reserved.

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

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

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

Table 81 – GID_PT request CT_IU

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

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

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

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

b) Bits 6-0 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 82 – Accept CT_IU to GID_PT request

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

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

The Name Server shall, 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 83. The requestor supplies the Fabric Port Name for which associated Port Identifiers are sought.

Table 83 – 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 84.

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 6-0 are reserved.

Table 84 – 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.

5.2.5.26 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 85. The requestor supplies the Port Identifier for which the Permanent Port Name object is sought.

Table 85 – 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 86.

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

Table 86 – Accept CT_IU to GPPN_ID request

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

5.2.5.27 Query - Get Port Identifiers (GID_FF)

The Name Server shall, 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 87. The requestor supplies the type code and FC-4 Feature bits for which supporting Port Identifiers are sought. The format of the FC-4 Feature bits field is:

a) Bits 7-4 are reserved; and

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

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

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

b) Bits 6 through 0 - reserved.

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

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

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

Table 87 – GID_FF request CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3
Flags	1
Domain_ID Scope	1
Area_ID Scope	1
reserved	1
reserved	2
FC-4 Feature bits	1
type code	1

The format of the Accept CT_IU to a GID_FF request is shown in table 88.

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

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

b) Bits 6-0 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 88 – 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 r)	1
Port Identifier #1	3
...	
Control (1 r r r r r r r r)	1
Port Identifier #n	3

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

The Name Server shall, 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 89. The requestor supplies the Domain_ID and Physical Port Number for which the Port Identifiers are sought.

Table 89 – 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 90. 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 6-0 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 90 – Accept CT_IU to GID_DP request

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

5.2.5.29 Register Node Name (RNN_ID)

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

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

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

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

The format of the RNN_ID request CT_IU is shown in table 91.

Table 91 – 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 92.

Table 92 – RNN_ID Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

5.2.5.30 Register Class of Service (RCS_ID)

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

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

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

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

The format of the RCS_ID request CT_IU is shown in table 93.

Table 93 – 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 94.

Table 94 – RCS_ID Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

5.2.5.31 Register FC-4 types (RFT_ID)

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

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

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

The format of the RFT_ID request CT_IU is shown in table 95.

Table 95 – 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 96.

Table 96 – RFT_ID Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

5.2.5.32 Register Symbolic Port Name (RSPN_ID)

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

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

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

The format of the RSPN_ID request CT_IU is shown in table 97. The Name length field shall contain a single byte unsigned value indicating the size of the variable length Symbolic Port Name.

Table 97 – 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 98.

Table 98 – RSPN_ID Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

5.2.5.33 Register Hard Address (RHA_ID)

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

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

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

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

The format of the RHA_ID request CT_IU is shown in table 99.

Table 99 – 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 100.

Table 100 – RHA_ID Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

5.2.5.34 Register FC-4 Features (RFF_ID)

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

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

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

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

The format of the RFF_ID request CT_IU is shown in table 101.

Table 101 – 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 7-4 are reserved; and

b) Bits 3-0 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 102.

Table 102 – RFF_ID Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

5.2.5.35 Register Symbolic Node Name (RSNN_NN)

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

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

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

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

The format of the RSNN_NN request CT_IU is shown in table 103. The Name length field shall contain a single byte unsigned value indicating the size of the variable length Symbolic Node Name.

Table 103 – 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 104.

Table 104 – RSNN_NN Accept CT_IU

Item	Size (Bytes)
CT_IU preamble	see 4.3

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

5.2.5.36 Remove all (DA_ID)

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

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

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

The format of the DA_ID request CT_IU is shown in table 105. The Port Identifier format shall be as defined in 5.2.3.2.

Table 105 – 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 106.

Table 106 – 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’.