Network Working Group
Request for Comments: 4171
Category: Standards Track

J. Tseng
Riverbed Technology
K. Gibbons
McDATA Corporation
F. Travostino
Nortel
C. Du Laney
Rincon Research Corporation
J. Souza
Microsoft

September 2005

Internet Storage Name Service (iSNS)

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2005).

Abstract

This document specifies the Internet Storage Name Service (iSNS) protocol, used for interaction between iSNS servers and iSNS clients, which facilitates automated discovery, management, and configuration of iSCSI and Fibre Channel devices (using iFCP gateways) on a TCP/IP network. iSNS provides intelligent storage discovery and management services comparable to those found in Fibre Channel networks, allowing a commodity IP network to function in a capacity similar to that of a storage area network. iSNS facilitates a seamless integration of IP and Fibre Channel networks due to its ability to emulate Fibre Channel fabric services and to manage both iSCSI and Fibre Channel devices. iSNS thereby provides value in any storage network comprised of iSCSI devices, Fibre Channel devices (using iFCP gateways), or any combination thereof.

Table of Contents

1. Introduction

1.1. Conventions Used in This Document

"iSNS" refers to the storage network model and associated services covered in the text of this document.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

All frame formats are in big endian network byte order.

All unused fields and bitmaps, including those that are RESERVED, SHOULD be set to zero when sending and ignored when receiving.

1.2. Purpose of This Document

This is a standards track document containing normative text specifying the iSNS Protocol, used by iSCSI and iFCP devices to communicate with the iSNS server. This document focuses on the interaction between iSNS servers and iSNS clients; interactions among multiple authoritative primary iSNS servers are a potential topic for future work.

2. iSNS Overview

iSNS facilitates scalable configuration and management of iSCSI and Fibre Channel (FCP) storage devices in an IP network by providing a set of services comparable to that available in Fibre Channel networks. iSNS thus allows a commodity IP network to function at a level of intelligence comparable to a Fibre Channel fabric. iSNS allows the administrator to go beyond a simple device-by-device management model, where each storage device is manually and individually configured with its own list of known initiators and targets. Using the iSNS, each storage device subordinates its discovery and management responsibilities to the iSNS server. The iSNS server thereby serves as the consolidated configuration point through which management stations can configure and manage the entire storage network, including both iSCSI and Fibre Channel devices.

iSNS can be implemented to support iSCSI and/or iFCP protocols as needed; an iSNS implementation MAY provide support for one or both of these protocols as desired by the implementor. Implementation requirements within each of these protocols are further discussed in Section 5. Use of iSNS is OPTIONAL for iSCSI and REQUIRED for iFCP.

2.1. iSNS Architectural Components

2.1.1. iSNS Protocol (iSNSP)

The iSNS Protocol (iSNSP) is a flexible and lightweight protocol that specifies how iSNS clients and servers communicate. It is suitable for various platforms, including switches and targets as well as server hosts.

2.1.2. iSNS Client

iSNS clients initiate transactions with iSNS servers using the iSNSP. iSNS clients are processes that are co-resident in the storage device, and that can register device attribute information, download information about other registered clients in a common Discovery Domain (DD), and receive asynchronous notification of events that occur in their DD(s). Management stations are a special type of iSNS client that have access to all DDs stored in the iSNS.

2.1.3. iSNS Server

iSNS servers respond to iSNS protocol queries and requests, and initiate iSNS protocol State Change Notifications. Properly authenticated information submitted by a registration request is stored in an iSNS database.

2.1.4. iSNS Database

The iSNS database is the information repository for the iSNS server(s). It maintains information about iSNS client attributes. A directory-enabled implementation of iSNS may store client attributes in an LDAP directory infrastructure.

2.1.5. iSCSI

iSCSI (Internet SCSI) is an encapsulation of SCSI for a new generation of storage devices interconnected with TCP/IP [iSCSI].

2.1.6. iFCP

iFCP (Internet FCP) is a gateway-to-gateway protocol designed to interconnect existing Fibre Channel and SCSI devices using TCP/IP. iFCP maps the existing FCP standard and associated Fibre Channel services to TCP/IP [iFCP].

2.2. iSNS Functional Overview

There are four main functions of the iSNS:

1) A Name Service Providing Storage Resource Discovery

2) Discovery Domain (DD) and Login Control Service

3) State Change Notification Service

4) Open Mapping of Fibre Channel and iSCSI Devices

2.2.1. Name Registration Service

The iSNS provides a registration function to allow all entities in a storage network to register and query the iSNS database. Both targets and initiators can register in the iSNS database, as well as query for information about other initiators and targets. This allows, for example, a client initiator to obtain information about target devices from the iSNS server. This service is modeled on the Fibre Channel Generic Services Name Server described in FC-GS-4, with extensions, operating within the context of an IP network.

The naming registration service also provides the ability to obtain a network-unique Domain ID for iFCP gateways when one is required.

2.2.2. Discovery Domain and Login Control Service

The Discovery Domain (DD) Service facilitates the partitioning of Storage Nodes into more manageable groupings for administrative and login control purposes. It allows the administrator to limit the login process of each host to the more appropriate subset of targets registered in the iSNS. This is particularly important for reducing the number of unnecessary logins (iSCSI logins or Fibre Channel Port Logins), and for limiting the amount of time that the host spends initializing login relationships as the size of the storage network scales up. Storage Nodes must be in at least one common enabled DD in order to obtain information about each other. Devices can be members of multiple DDs simultaneously.

Login Control allows targets to delegate their access control/authorization policies to the iSNS server. This is consistent with the goal of centralizing management of those storage devices using the iSNS server. The target node or device downloads the list of authorized initiators from the iSNS. Each node or device is uniquely identified by an iSCSI Name or FC Port Name. Only

initiators that match the required identification and authorization provided by the iSNS will be allowed access by that target Node during session establishment.

Placing Portals of a Network Entity into Discovery Domains allows administrators to indicate the preferred IP Portal interface through which storage traffic should access specific Storage Nodes of that Network Entity. If no Portals of a Network Entity have been placed into a DD, then queries scoped to that DD SHALL report all Portals of that Network Entity. If one or more Portals of a Network Entity have been placed into a DD, then queries scoped to that DD SHALL report only those Portals that have been explicitly placed in the DD.

DDs can be managed offline through a separate management workstation using the iSNSP or SNMP. If the target opts to use the Login Control feature of the iSNS, the target delegates management of access control policy (i.e., the list of initiators allowed to log in to that target) to the management workstations that are managing the configuration in the iSNS database.

If administratively authorized, a target can upload its own Login Control list. This is accomplished using the DDReg message and listing the iSCSI name of each initiator to be registered in the target's DD.

An implementation MAY decide that newly registered devices that have not explicitly been placed into a DD by the management station will be placed into a "default DD" contained in a "default DDS" whose initial DD Set Status value is "enabled". This makes them visible to other devices in the default DD. Other implementations MAY decide that they are registered with no DD, making them inaccessible to source-scoped iSNSP messages.

The iSNS server uses the Source Attribute of each iSNSP message to determine the originator of the request and to scope the operation to a set of Discovery Domains. In addition, the Node Type (specified in the iFCP or iSCSI Node Type bitmap field) may also be used to determine authorization for the specified iSNS operation. For example, only Control Nodes are authorized to create or delete discovery domains.

Valid and active Discovery Domains (DDs) belong to at least one active Discovery Domain Set (DDS). Discovery Domains that do not belong to an activated DDS are not enabled. The iSNS server MUST maintain the state of DD membership for all Storage Nodes, even for those that have been deregistered. DD membership is persistent regardless of whether a Storage Node is actively registered in the iSNS database.

2.2.3. State Change Notification Service

The State Change Notification (SCN) service allows the iSNS Server to issue notifications about network events that affect the operational state of Storage Nodes. The iSNS client may register for notifications on behalf of its Storage Nodes for notification of events detected by the iSNS Server. SCNs notify iSNS clients of explicit or implicit changes to the iSNS database; they do not necessarily indicate the state of connectivity to peer storage devices in the network. The response of a storage device to receipt of an SCN is implementation-specific; the policy for responding to SCNs is outside of the scope of this document.

There are two types of SCN registrations: regular registrations and management registrations. Management registrations result in management SCNs, whereas regular registrations result in regular SCNs. The type of registration and SCN message is indicated in the SCN bitmap (see Sections 6.4.4 and 6.6.12).

A regular SCN registration indicates that the Discovery Domain Service SHALL be used to control the distribution of SCN messages. Receipt of regular SCNs is limited to the discovery domains in which the SCN-triggering event takes place. Regular SCNs do not contain information about discovery domains.

A management SCN registration can only by requested by Control Nodes. Management SCNs resulting from management registrations are not bound by the Discovery Domain service. Authorization to request management SCN registrations may be administratively controlled.

The iSNS server SHOULD be implemented with hardware and software resources sufficient to support the expected number of iSNS clients. However, if resources are unexpectedly exhausted, then the iSNS server MAY refuse SCN service by returning an SCN Registration Rejected (Status Code 17). The rejection might occur in situations where the network size or current number of SCN registrations has passed an implementation-specific threshold. A client not allowed to register for SCNs may decide to monitor its sessions with other storage devices directly.

The specific notification mechanism by which the iSNS server learns of the events that trigger SCNs is implementation-specific, but can include examples such as explicit notification messages from an iSNS client to the iSNS server, or a hardware interrupt to a switch-hosted iSNS server as a result of link failure.

2.2.4. Open Mapping between Fibre Channel and iSCSI Devices

The iSNS database stores naming and discovery information about both Fibre Channel and iSCSI devices. This allows the iSNS server to store mappings of a Fibre Channel device to a proxy iSCSI device "image" in the IP network. Similarly, mappings of an iSCSI device to a "proxy WWN" can be stored under the WWNN Token field for that iSCSI device.

Furthermore, through use of iSCSI-FC gateways, Fibre Channel-aware management stations can interact with the iSNS server to retrieve information about Fibre Channel devices, and use this information to manage Fibre Channel and iSCSI devices. This allows management functions such as Discovery Domains and State Change Notifications to be applied seamlessly to both iSCSI and Fibre Channel devices, facilitating integration of IP networks with Fibre Channel devices and fabrics.

Note that Fibre Channel attributes are stored as iFCP attributes, and that the ability to store this information in the iSNS server is useful even if the iFCP protocol is not implemented. In particular, tag 101 can be used to store a "Proxy iSCSI Name" for Fibre Channel devices registered in the iSNS server. This field is used to associate the FC device with an iSCSI registration entry that is used for the Fibre Channel device to communicate with iSCSI devices in the IP network. Conversely, tag 37 (see Section 6.1) contains a WWNN Token field, which can be used to store an FC Node Name (WWNN) value used by iSCSI-FC gateways to represent an iSCSI device in the Fibre Channel domain.

By storing the mapping between Fibre Channel and iSCSI devices in the iSNS server, this information becomes open to any authorized iSNS client wishing to retrieve and use this information. In many cases, this provides advantages over storing the information internally within an iSCSI-FC gateway, where the mapping is inaccessible to other devices except by proprietary mechanisms.

2.3. iSNS Usage Model

The following is a high-level description of how each type of device in a storage network can utilize iSNS. Each type of device interacts with the iSNS server as an iSNS client and must register itself in the iSNS database in order to access services provided by the iSNS.

2.3.1. iSCSI Initiator

An iSCSI initiator will query the iSNS server to discover the presence and location of iSCSI target devices. It may also request state change notifications (SCNs) so that it can be notified of new targets that appear on the network after the initial bootup and discovery. SCNs can also inform the iSCSI initiator of targets that have been removed from or no longer available in the storage network, so that incomplete storage sessions can be gracefully terminated and resources for non-existent targets can be reallocated.

2.3.2. iSCSI Target

An iSCSI target allows itself to be discovered by iSCSI initiators by registering its presence in the iSNS server. It may also register for SCNs in order to detect the addition or removal of initiators for resource allocation purposes. The iSCSI target device may also register for Entity Status Inquiry (ESI) messages, which allow the iSNS to monitor the target device’s availability in the storage network.

2.3.3. iSCSI-FC Gateway

An iSCSI-FC gateway bridges devices in a Fibre Channel network to an iSCSI/IP network. It may use the iSNS server to store FC device attributes discovered in the FC name server, as well as mappings of FC device identifiers to iSCSI device identifiers. iSNS has the capability to store all attributes of both iSCSI and Fibre Channel devices; iSCSI devices are managed through direct interaction using iSNS, while FC devices can be indirectly managed through iSNS interactions with the iSCSI-FC gateway. This allows both iSCSI and Fibre Channel devices to be managed in a seamless management framework.

2.3.4. iFCP Gateway

An iFCP gateway uses iSNS to emulate the services provided by a Fibre Channel name server for FC devices in its gateway region. iSNS provides basic discovery and zoning configuration information to be enforced by the iFCP gateway. When queried, iSNS returns information on the N_Port network address used to establish iFCP sessions between FC devices supported by iFCP gateways.

2.3.5. Management Station

A management station uses iSNS to monitor storage devices and to enable or disable storage sessions by configuring discovery domains. A management station usually interacts with the iSNS server as a

Control Node endowed with access to all iSNS database records and with special privileges to configure discovery domains. Through manipulation of discovery domains, the management station controls the scope of device discovery for iSNS clients querying the iSNS server.

2.4. Administratively Controlled iSNS Settings

Some important operational settings for the iSNS server are configured using administrative means, such as a configuration file, a console port, an SNMP, or another implementation-specific method. These administratively-controlled settings cannot be configured using the iSNS Protocol, and therefore the iSNS server implementation MUST provide for such an administrative control interface.

The following is a list of parameters that are administratively controlled for the iSNS server. In the absence of alternative settings provided by the administrator, the following specified default settings MUST be used.

Setting	Default Setting
ESI Non-Response Threshold	3 (see 5.6.5.13)
Management SCNs (Control Nodes only)	enabled (see 5.6.5.8)
Default DD/DDS	disabled
DD/DDS Modification	
- Control Node	enabled
- iSCSI Target Node Type	disabled
- iSCSI Initiator Node Type	disabled
- iFCP Target Port Role	disabled
- iFCP Initiator Port Role	disabled
Authorized Control Nodes	N/A

ESI Non-Response Threshold: determines the number of ESI messages sent without receiving a response before the network entity is deregistered from the iSNS database.

Management SCN for Control Node: determines whether a registered Control Node is permitted to register to receive Management SCNs.

Default DD/DDS: determines whether a newly registered device not explicitly placed into a discovery domain (DD) and discovery domain set (DDS) is placed into a default DD/DDS.

DD/DDS Modification: determines whether the specified type of Node is allowed to add, delete or update DDs and DDSs.

Authorized Control Nodes: a list of Nodes identified by iSCSI Name or FC Port Name WWPN that are authorized to register as Control Nodes.

2.5. iSNS Server Discovery

2.5.1. Service Location Protocol (SLP)

The Service Location Protocol (SLP) provides a flexible and scalable framework for providing hosts with access to information about the existence, location, and configuration of networked services, including the iSNS server. SLP can be used by iSNS clients to discover the IP address or FQDN of the iSNS server. To implement discovery through SLP, a Service Agent (SA) should be cohosted in the iSNS server, and a User Agent (UA) should be in each iSNS client. Each client multicasts a discovery message requesting the IP address of the iSNS server(s). The SA responds to this request. Optionally, the location of the iSNS server can be stored in the SLP Directory Agent (DA).

Note that a complete description and specification of SLP can be found in [RFC2608], and is beyond the scope of this document. A service template for using SLP to locate iSNS servers can be found in [iSCSI-SLP].

2.5.2. Dynamic Host Configuration Protocol (DHCP)

The IP address of the iSNS server can be stored in a DHCP server to be downloaded by iSNS clients using a DHCP option. The DHCP option number to be used for distributing the iSNS server location is found in [iSNSOption].

2.5.3. iSNS Heartbeat Message

The iSNS heartbeat message is described in Section 5.6.5.14. It allows iSNS clients within the broadcast or multicast domain of the iSNS server to discover the location of the active iSNS server and any backup servers.

2.6. iSNS and Network Address Translation (NAT)

The existence of NAT will have an impact upon information retrieved from the iSNS server. If the iSNS client exists in an addressing domain different from that of the iSNS server, then IP address information stored in the iSNS server may not be correct when interpreted in the domain of the iSNS client.

There are several possible approaches to allow operation of iSNS within a NAT network. The first approach is to require use of the canonical TCP port number by both targets and initiators when addressing targets across a NAT boundary, and for the iSNS client not to query for nominal IP addresses. Rather, the iSNS client queries for the DNS Fully Qualified Domain Name stored in the Entity Identifier field when seeking addressing information. Once retrieved, the DNS name can be interpreted in each address domain and mapped to the appropriate IP address by local DNS servers.

A second approach is to deploy a distributed network of iSNS servers. Local iSNS servers are deployed inside and outside NAT boundaries, with each local server storing relevant IP addresses for their respective NAT domains. Updates among the network of decentralized, local iSNS servers are handled using LDAP and appropriate NAT translation rules implemented within the update mechanism in each server.

Finally, note that it is possible for an iSNS server in the private addressing domain behind a NAT boundary to exclusively support iSNS clients that are operating in the global IP addressing domain. If this is the case, the administrator only needs to ensure that the appropriate mappings are configured on the NAT gateways to allow the iSNS clients to initiate iSNSP sessions to the iSNS server. All registered addresses contained in the iSNS server are thus public IP addresses for use outside the NAT boundary. Care should be taken to ensure that there are no iSNS clients querying the server from inside the NAT boundary.

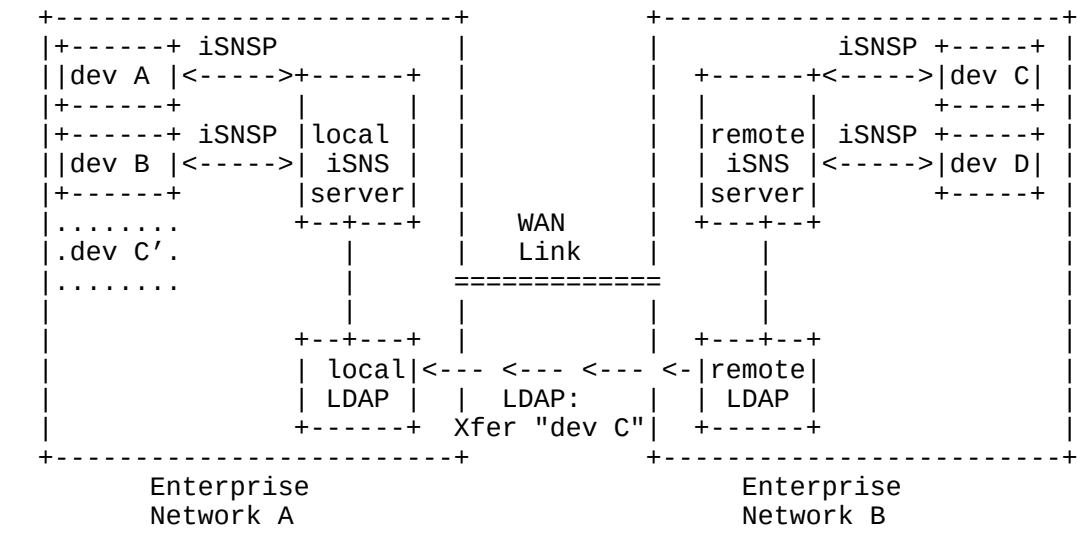
2.7. Transfer of iSNS Database Records between iSNS Servers

Transfer of iSNS database records between iSNS servers has important applications, including the following:

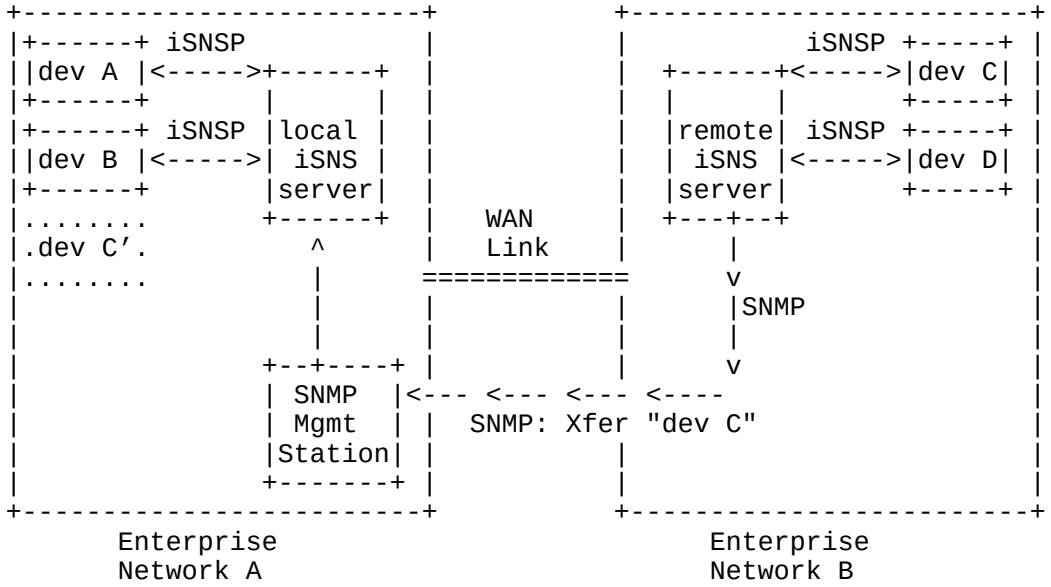
1) An independent organization needs to transfer storage information to a different organization. Each organization independently maintains its own iSNS infrastructure. To facilitate discovery of storage assets of the peer organization using IP, iSNS database records can be transferred between authoritative iSNS servers from each organization. This allows storage sessions to be established directly between devices residing in each organization’s storage network infrastructure over a common IP network.

2) Multiple iSNS servers are desired for redundancy. Backup servers need to maintain copies of the primary server’s dynamically changing database.

To support the above applications, information in an iSNS server can be distributed to other iSNS servers either using the iSNS protocol, or through out-of-band mechanisms using non-iSNS protocols. The following examples illustrate possible methods for transferring data records between iSNS servers. In the first example, a back-end LDAP information base is used to support the iSNS server, and the data is transferred using the LDAP protocol. Once the record transfer of the remote device is completed, it becomes visible and accessible to local devices using the local iSNS server. This allows local devices to establish sessions with remote devices (provided that firewall boundaries can be negotiated).



In the above diagram, two business partners wish to share storage "dev C". Using LDAP, the record for "dev C" can be transferred from Network B to Network A. Once accessible to the local iSNS server in Network A, local devices A and B can now discover and connect to "dev C".



The above diagram illustrates a second example of how iSNS records can be shared. This method uses an SNMP-based management station to retrieve (GET) the desired record for "dev C" manually, and then to store (SET) it on the local iSNS server directly. Once the record is transferred to the local iSNS server in Network A, "dev C" becomes visible and accessible (provided that firewall boundaries can be negotiated) to other devices in Network A.

Other methods, including proprietary protocols, can be used to transfer device records between iSNS servers. Further discussion and explanation of these methodologies is beyond the scope of this document.

2.8. Backup iSNS Servers

This section offers a broad framework for implementation and deployment of iSNS backup servers. Server failover and recovery are topics of continuing research, and adequate resolution of issues such as split brain and primary server selection is dependent on the specific implementation requirements and deployment needs. The failover mechanisms discussed in this document focus on the interaction between iSNS clients and iSNS servers. Specifically, what is covered in this document includes the following:

- iSNS client behavior and the iSNS protocol interaction between the client and multiple iSNS servers, some of which are backup servers.

- Required failover behaviors of the collection of iSNS servers that includes active and backup servers.

However, note that this document does not specify the complete functional failover requirements of each iSNS server. In particular, it does not specify the complete set of protocol interactions among the iSNS servers that are required to achieve stable failover operation in an interoperable manner.

For the purposes of this discussion, the specified backup mechanisms pertain to interaction among different logical iSNS servers. Note that it is possible to create multiple physical iSNS servers to form a single logical iSNS server cluster, and thus to distribute iSNS transaction processing among multiple physical servers. However, a more detailed discussion of the interactions between physical servers within a logical iSNS server cluster is beyond the scope of this document.

Multiple logical iSNS servers can be used to provide redundancy in the event that the active iSNS server fails or is removed from the network. The methods described in Section 2.7 above can be used to transfer name server records to backup iSNS servers. Each backup server maintains a redundant copy of the name server database found in the primary iSNS server, and can respond to iSNS protocol messages in the same way as the active server. Each backup server SHOULD monitor the health and status of the active iSNS server, including checking to make sure its own database is synchronized with the active server's database. How each backup server accomplishes this is implementation-dependent, and may (or may not) include using the iSNS protocol. If the iSNS protocol is used, then the backup server MAY register itself in the active server's iSNS database as a Control Node, allowing it to receive state-change notifications.

Generally, the administrator or some automated election process is responsible for initial and subsequent designation of the primary server and each backup server.

A maximum of one logical backup iSNS server SHALL exist at any individual IP address, in order to avoid conflicts from multiple servers listening on the same canonical iSNS TCP or UDP port number.

The iSNS heartbeat can also be used to coordinate the designation and selection of primary and backup iSNS servers.

Each backup server MUST note its relative precedence in the active server's list of backup servers. If its precedence is not already known, each backup server MAY learn it from the iSNS heartbeat message, by noting the position of its IP address in the ordered list

of backup server IP addresses. For example, if it is the first backup listed in the heartbeat message, then its backup precedence is 1. If it is the third backup server listed, then its backup precedence is 3.

If a backup server establishes that it has lost connectivity to the active server and other backup servers of higher precedence, then it SHOULD assume that it is the active server. The method of determining whether connectivity has been lost is implementation-specific. One possible approach is to assume that if the backup server does not receive iSNS heartbeat messages for a period of time, then connectivity to the active server has been lost. Alternatively, the backup server may establish TCP connections to the active server and other backup servers, with loss of connectivity determined through non-response to periodic echo or polling messages (using iSNSP, SNMP, or other protocols).

When a backup server becomes the active server, it SHALL assume all active server responsibilities, including (if used) transmission of the iSNS heartbeat message. If transmitting the iSNS heartbeat, the backup server replaces the active Server IP Address and TCP/UDP Port entries with its own IP address and TCP/UDP Port, and begins incrementing the counter field from the last known value from the previously-active iSNS server. However, it MUST NOT change the original ordered list of backup server IP Address and TCP/UDP Port entries. If the primary backup server or other higher-precedence backup server returns, then the existing active server is responsible for ensuring that the new active server's database is up-to-date before demoting itself to its original status as backup.

Since the primary and backup iSNS servers maintain a coordinated database, no re-registration by an iSNS Client is required when a backup server takes the active server role. Likewise, no re-registration by an iSNS Client is required when the previous primary server returns to the active server role.

2.9. Transport Protocols

The iSNS Protocol is transport-neutral. Query and registration messages are transported over TCP or UDP. iSNS heartbeat messages are transported using IP multicast or broadcast.

2.9.1. Use of TCP for iSNS Communication

It MUST be possible to use TCP for iSNS communication. The iSNS server MUST accept TCP connections for client registrations. To receive Entity Status Inquiry (ESI) (see Section 5.6.5.13) monitoring the use of TCP, the client registers the Portal ESI Interval and the

port number of the TCP port that will be used to receive ESI messages. The iSNS server initiates the TCP connection used to deliver the ESI message. This TCP connection does not need to be continuously open.

To receive SCN notifications using TCP, the client registers the iSCSI or iFCP SCN Bitmap and the port number of the TCP port in the Portal used to receive SCNs. The iSNS server initiates the TCP connection used to deliver the SCN message. This TCP connection does not need to be continuously open.

It is possible for an iSNS client to use the same TCP connection for SCN, ESI, and iSNS queries. Alternatively, separate connections may be used.

2.9.2. Use of UDP for iSNS Communication

The iSNS server MAY accept UDP messages for client registrations. The iSNS server MUST accept registrations from clients requesting UDP-based ESI and SCN messages.

To receive UDP-based ESI monitoring messages, the client registers the port number of the UDP port in at least one Portal to be used to receive and respond to ESI messages from the iSNS server. If a Network Entity has multiple Portals with registered ESI UDP Ports, then ESI messages SHALL be delivered to every Portal registered to receive such messages.

To receive UDP-based SCN notification messages, the client registers the port number of the UDP port in at least one Portal to be used to receive SCN messages from the iSNS server. If a Network Entity has multiple Portals with registered SCN UDP Ports, then SCN messages SHALL be delivered to each Portal registered to receive such messages.

When using UDP to transport iSNS messages, each UDP datagram MUST contain exactly one iSNS PDU (see Section 5).

2.9.3. iSNS Multicast and Broadcast Messages

iSNS multicast messages are transported using IP multicast or broadcast. The iSNS heartbeat is the only iSNS multicast or broadcast message. This message is originated by the iSNS server and sent to all iSNS clients that are listening on the IP multicast address allocated for the iSNS heartbeat.

2.10. Simple Network Management Protocol (SNMP) Requirements

The iSNS Server may be managed via the iSNS MIB [iSNSMIB] using an SNMP management framework [RFC3411]. For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to Section 7 of RFC 3410 [RFC3410]. The iSNS MIB provides the ability to configure and monitor an iSNS server without using the iSNS protocol directly. SNMP management frameworks have several requirements for object indexing in order for objects to be accessed or added.

SNMP uses an Object Identifier (OID) for object identification. The size of each OID is restricted to a maximum of 128 sub-identifiers. Both the iSCSI and iFCP protocol contain identifiers, such as the iSCSI Name, that are greater the 128 characters in length. Using such identifiers as an index would result in more than 128 sub-identifiers per OID. In order to support objects that have key identifiers whose maximum length is longer than the maximum SNMP-supported length, the iSNS server provides secondary non-zero integer index identifiers. These indexes SHALL be persistent for as long as the server is active. Furthermore, index values for recently deregistered objects SHOULD NOT be reused in the short term. Object attributes, including indexes, are described in detail in Section 6.

For SNMP based management applications to create a new entry in a table of objects, a valid OID must be available to specify the table row. The iSNS server supports this by providing, for each type of object that can be added via SNMP, an object attribute that returns the next available non-zero integer index. This allows an SNMP client to request an OID to be used for registering a new object in the server. Object attributes, including next available index attributes, are described in detail in Section 6.

3. iSNS Object Model

iSNS provides the framework for the registration, discovery, and management of iSCSI devices and Fibre Channel-based devices (using iFCP). This architecture framework provides elements needed to describe various storage device objects and attributes that may exist on an IP storage network. Objects defined in this architecture framework include Network Entity, Portal, Storage Node, FC Device, Discovery Domain, and Discovery Domain Set. Each of these objects is described in greater detail in the following sections.

3.1. Network Entity Object

The Network Entity object is a container of Storage Node objects and Portal objects. It represents the infrastructure supporting access to a unique set of one or more Storage Nodes. The Entity Identifier attribute uniquely distinguishes a Network Entity, and is the key used to register a Network Entity object in an iSNS server. All Storage Nodes and Portals contained within a single Network Entity object operate as a cohesive unit.

Note that it is possible for a single physical device or gateway to be represented by more than one logical Network Entity in the iSNS database. For example, one of the Storage Nodes on a physical device may be accessible from only a subset of the network interfaces (i.e., Portals) available on that device. In this case, a logical network entity (i.e., a "shadow entity") is created and used to contain the Portals and Storage Nodes that can operate cooperatively. No object (Portals, Storage Nodes, etc.) can be contained in more than one logical Network Entity.

Similarly, it is possible for a logical Network Entity to be supported by more than one physical device or gateway. For example, multiple FC-iSCSI gateways may be used to bridge FC devices in a single Fibre Channel network. Collectively, the multiple gateways can be used to support a single logical Network Entity that is used to contain all the devices in that Fibre Channel network.

3.2. Portal Object

The Portal object is an interface through which access to Storage Nodes within the Network Entity can be obtained. The IP address and TCP/UDP Port number attributes uniquely distinguish a Portal object, and combined are the key used to register a Portal object in an iSNS server. A Portal is contained in one and only one Network Entity, and may be contained in one or more DDs (see Section 3.6).

3.3. Storage Node Object

The Storage Node object is the logical endpoint of an iSCSI or iFCP session. In iFCP, the session endpoint is represented by the World Wide Port Name (WWPN). In iSCSI, the session endpoint is represented by the iSCSI Name of the device. For iSCSI, the iSCSI Name attribute uniquely distinguishes a Storage Node, and is the key used to register a Storage Node object in an iSNS Server. For iFCP, the FC Port Name (WWPN) attribute uniquely distinguishes a Storage Node, and is the key used to register a Storage Node object in the iSNS Server. Storage Node is contained in only one Network Entity object and may be contained in one or more DDs (see Section 3.6).

3.4. Portal Group Object

The Portal Group (PG) object represents an association between a Portal and an iSCSI Node. Each Portal and iSCSI Storage Node registered in an Entity can be associated using a Portal Group (PG) object. The PG Tag (PGT), if non-NULL, indicates that the associated Portal provides access to the associated iSCSI Storage Node in the Entity. All Portals that have the same PGT value for a specific iSCSI Storage Node allow coordinated access to that node.

A PG object MAY be registered when a Portal or iSCSI Storage Node is registered. Each Portal to iSCSI Node association is represented by one and only one PG object. In order for a Portal to provide access to an iSCSI Node, the PGT of the PG object MUST be non-NULL. If the PGT value registered for a specified Portal and iSCSI Node is NULL, or if no PGT value is registered, then the Portal does not provide access to that iSCSI Node in the Entity.

The PGT value indicates whether access to an iSCSI Node can be coordinated across multiple Portals. All Portals that have the same PGT value for a specific iSCSI Node can provide coordinated access to that iSCSI Node. According to the iSCSI Specification, coordinated access to an iSCSI node indicates the capability of coordinating an iSCSI session with connections that span these Portals [iSCSI].

The PG object is uniquely distinguished by the iSCSI Name, Portal IP Address, and Portal TCP Port values of the associated Storage Node and Portal objects. These are represented in the iSNS Server by the PG iSCSI Name, PG Portal IP Address, and PG Portal TCP/UDP Port attributes, respectively. The PG object is also uniquely distinguished in the iSNS Server by the PG Index value.

A new PG object can only be registered by referencing its associated iSCSI Storage Node or Portal object. A pre-existing PG object can be modified or queried by using its Portal Group Index as message key, or by referencing its associated iSCSI Storage Node or Portal object. A 0-length Tag, Length, Value TLV is used to register a PGT NULL value.

The PG object is deregistered if and only if its associated iSCSI Node and Portal objects are both removed.

3.5. Device Object

The FC Device represents the Fibre Channel Node. This object contains information that may be useful in the management of the Fibre Channel device. The FC Node Name (WWNN) attribute uniquely distinguishes an FC Device, and is the key used to register an FC Device object in the iSNS Server.

The FC Device is contained in one or more Storage Node objects.

3.6. Discovery Domain Object

Discovery Domains (DD) are a security and management mechanism used to administer access and connectivity to storage devices. For query and registration purposes, they are considered containers for Storage Node and Portal objects. A query by an iSNS client that is not from a Control Node only returns information about objects with which it shares at least one active DD. The only exception to this rule is with Portals; if Storage Nodes of a Network Entity are registered in the DD without Portals, then all Portals of that Network Entity are implicit members of that DD. The Discovery Domain ID (DD_ID) attribute uniquely distinguishes a Discovery Domain object, and is the key used to register a Discovery Domain object in the iSNS Server.

A DD is considered active if it is a member of at least one active DD Set. DDs that are not members of at least one enabled DDS are considered disabled. A Storage Node can be a member of one or more DDs. An enabled DD establishes connectivity among the Storage Nodes in that DD.

3.7. Discovery Domain Set Object

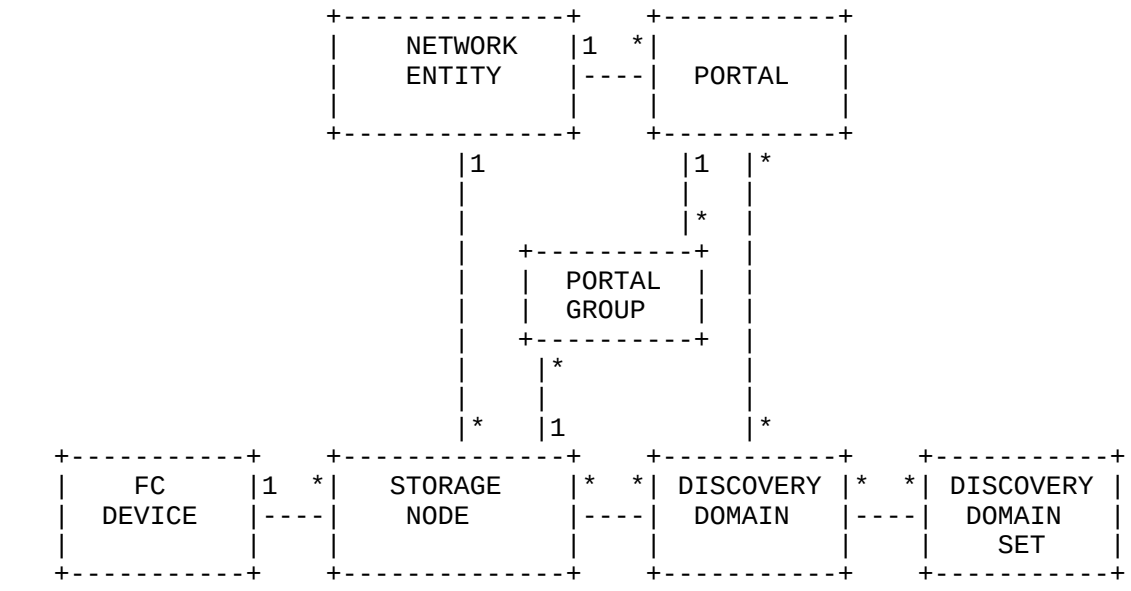
The Discovery Domain Set (DDS) is a container object for Discovery Domains (DDs). DDSs may contain one or more DDs. Similarly, each DD can be a member of one or more DDSs. DDSs are a mechanism to store coordinated sets of DD mappings in the iSNS server. Active DDs are members of at least one active DD Set. Multiple DDSs may be considered active at the same time. The Discovery Domain Set ID (DDS_ID) attribute uniquely distinguishes a Discovery Domain Set object, and is the key used to register a Discovery Domain Set object in the iSNS Server.

3.8. Database Model

As presented to the iSNS client, each object of a specific type in the iSNS database MUST have an implicit internal linear ordering based on the key(s) for that object type. This ordering provides the

ability to respond to DevGetNext queries (see Section 5.6.5.3). The ordering of objects in the iSNS database SHOULD NOT be changed with respect to that implied ordering, as a consequence of object insertions and deletions. That is, the relative order of surviving object entries in the iSNS database SHOULD be preserved so that the DevGetNext message encounters generally reasonable behavior.

The following diagram shows the various objects described above and their relationship to each other.



* represents 0 to many possible relationships

4. iSNS Implementation Requirements

This section details specific requirements for support of each of these IP storage protocols. Implementation requirements for security are described in Section 7.

4.1. iSCSI Requirements

Use of iSNS in support of iSCSI is OPTIONAL. iSCSI devices MAY be manually configured with the iSCSI Name and IP address of peer devices, without the aid or intervention of iSNS. iSCSI devices may also use SLP [RFC2608] to discover peer iSCSI devices. However, iSNS is useful for scaling a storage network to a larger number of iSCSI devices.

4.1.1. Required Attributes for Support of iSCSI

The following attributes are available to support iSCSI. Attributes indicated in the REQUIRED for Server column MUST be implemented by an iSNS server used to support iSCSI. Attributes indicated in the REQUIRED for Client column MUST be implemented by an iSCSI device that elects to use the iSNS. Attributes indicated in the K (Key) column uniquely identify the object type in the iSNS Server. A more detailed description of each attribute is found in Section 6.

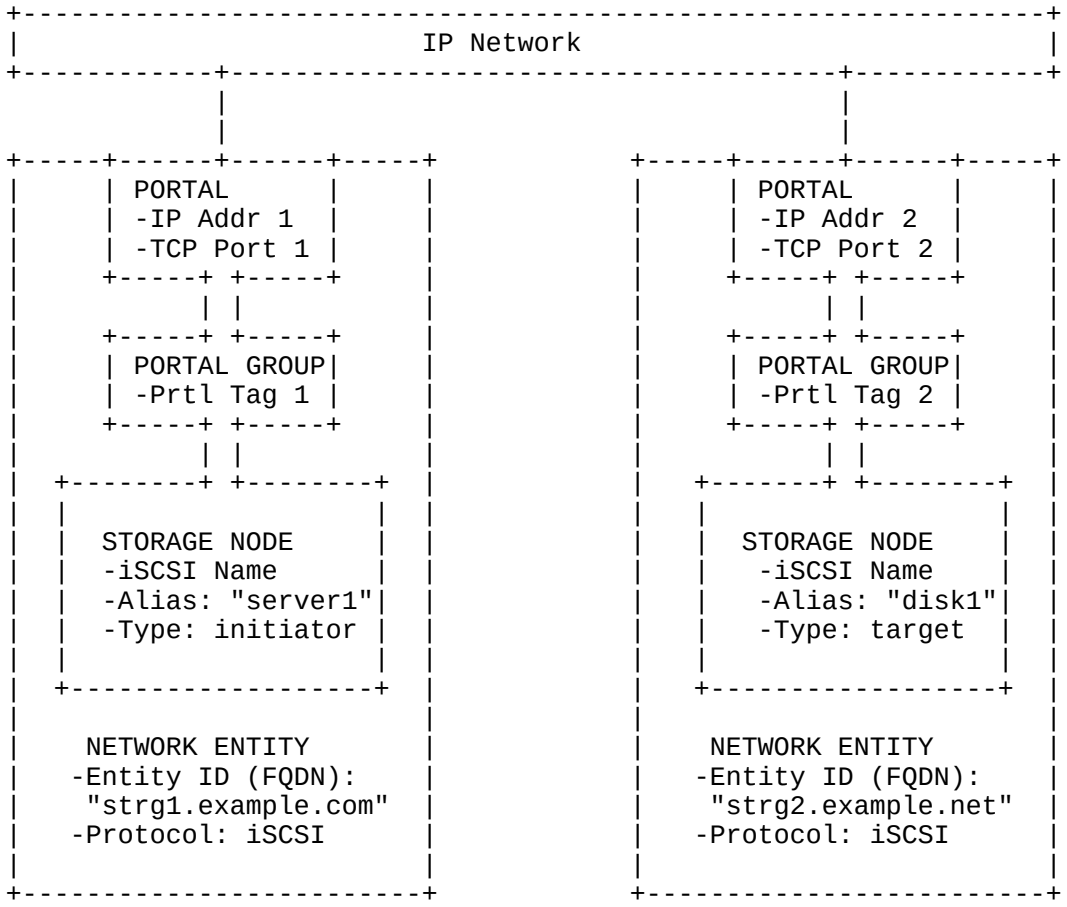
			REQUIRED for:	
Object	Attribute	K	Server	Client
NETWORK ENTITY	Entity Identifier	*	*	*
	Entity Protocol		*	*
	Management IP Address		*	
	Timestamp		*	
	Protocol Version Range		*	
	Registration Period		*	
	Entity Index		*	
	Entity IKE Phase-1 Proposal			
	Entity Certificate			
PORTAL	IP Address	*	*	*
	TCP/UDP Port	*	*	*
	Portal Symbolic Name		*	
	ESI Interval		*	
	ESI Port		*	
	Portal Index		*	
	SCN Port		*	
	Portal Security Bitmap		*	
	Portal IKE Phase-1 Proposal			
	Portal IKE Phase-2 Proposal			
	Portal Certificate			
PORTAL GROUP	PG iSCSI Name	*	*	*
	PG IP Address	*	*	*
	PG TCP/UDP Port	*	*	*
	PG Tag		*	*
	PG Index		*	

STORAGE NODE	iSCSI Name	*	*	*
	iSCSI Node Type		*	*
	Alias		*	
	iSCSI SCN Bitmap		*	
	iSCSI Node Index		*	
	WWNN Token			
	iSCSI AuthMethod			
	iSCSI Node Certificate			
DISCOVERY DOMAIN	DD ID	*	*	*
	DD Symbolic Name		*	
	DD Member iSCSI Node Index		*	
	DD Member iSCSI Name		*	
	DD Member Portal Index		*	
	DD Member Portal IP Addr		*	
	DD Member Portal TCP/UDP		*	
	DD Features		*	
DISCOVERY DOMAIN SET	DDS Identifier	*	*	
	DDS Symbolic Name		*	
	DDS Status		*	

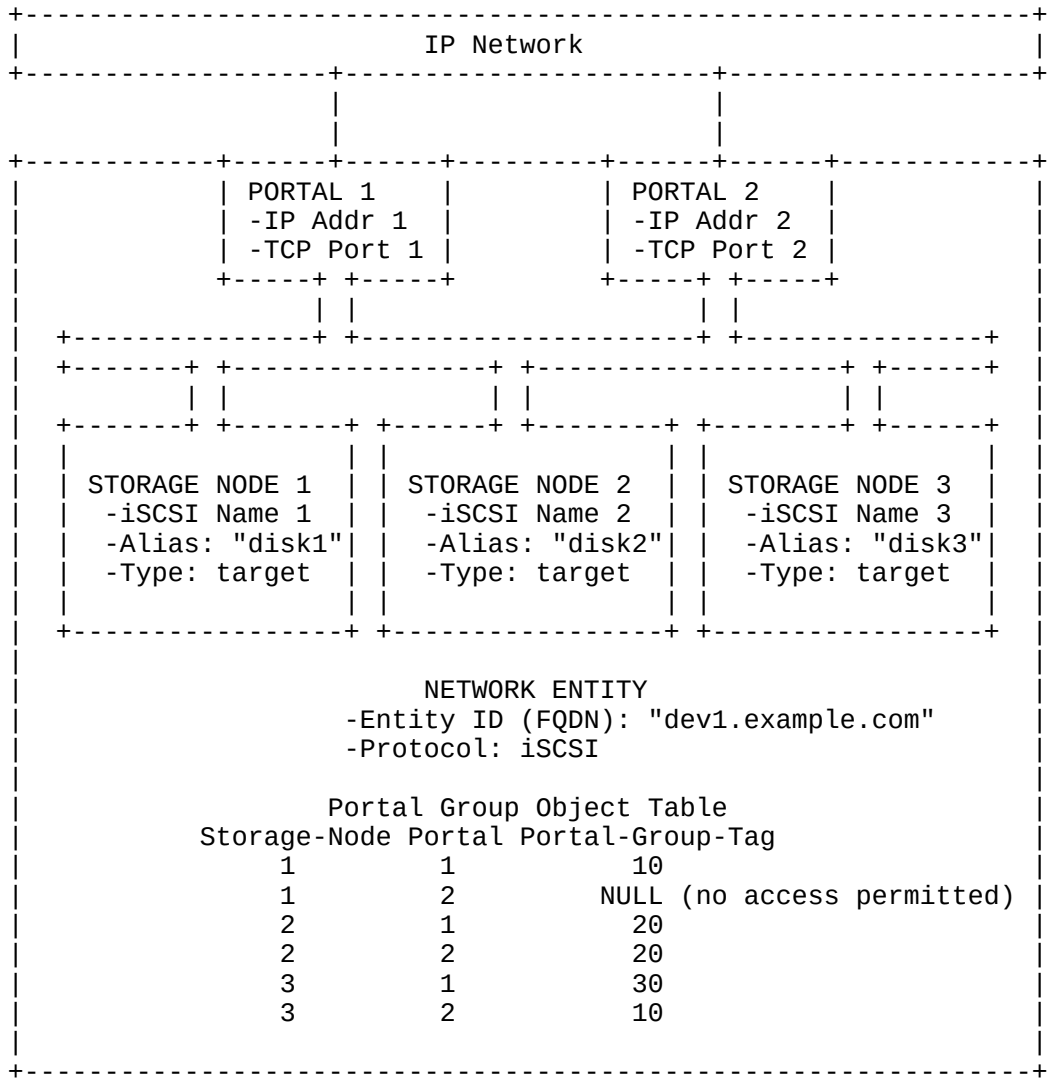
All iSCSI user-specified and vendor-specified attributes are OPTIONAL to implement and use.

4.1.2. Examples: iSCSI Object Model Diagrams

The following diagram models how a simple iSCSI-based initiator and target is represented using database objects stored in the iSNS server. In this implementation, each target and initiator is attached to a single Portal.



The object model can be expanded to describe more complex devices, such as an iSCSI device with more than one storage controller, in which each controller is accessible through any of multiple Portal interfaces, possibly using multiple Portal Groups. The storage controllers on this device can be accessed through alternate Portal interfaces if any original interface should fail. The following diagram describes such a device:



Storage Node 1 is accessible via Portal 1 with a PGT of 10. It does not have a Portal Group Tag (PGT) assigned for Portal 2, so Storage Node 1 cannot be accessed via Portal 2.

Storage Node 2 can be accessed via both Portal 1 and Portal 2. Since Storage Node 2 has the same PGT value assigned to both Portal 1 and Portal 2, in this case 20, coordinated access via the Portals is available [iSCSI].

Storage Node 3 can be accessed via Portal 1 or Portal 2. However, since Storage Node 3 has different PGT values assigned to each Portal, in this case 10 and 30, access is not coordinated [iSCSI]. Because PGTs are assigned within the context of a Storage Node, the PGT value of 10 used for Storage Node 1 and Storage Node 3 are not interrelated.

4.1.3. Required Commands and Response Messages for Support of iSCSI

The following iSNSP messages and responses are available in support of iSCSI. Messages indicated in the REQUIRED for Server column MUST be implemented in iSNS servers used for iSCSI devices. Messages indicated in the REQUIRED for Client column MUST be implemented in iSCSI devices that elect to use the iSNS server.

Message Description	Abbreviation	Func_ID	REQUIRED for: Server	Client
RESERVED		0x0000		
Device Attr Reg Request	DevAttrReg	0x0001	*	*
Dev Attr Query Request	DevAttrQry	0x0002	*	*
Dev Get Next Request	DevGetNext	0x0003	*	
Deregister Dev Request	DevDereg	0x0004	*	*
SCN Register Request	SCNReg	0x0005	*	
SCN Deregister Request	SCNDereg	0x0006	*	
SCN Event	SCNEvent	0x0007	*	
State Change Notification	SCN	0x0008	*	
DD Register	DDReg	0x0009	*	*
DD Deregister	DDDereg	0x000A	*	*
DDS Register	DDSReg	0x000B	*	*
DDS Deregister	DDSDereg	0x000C	*	*
Entity Status Inquiry	ESI	0x000D	*	
Name Service Heartbeat	Heartbeat	0x000E		
RESERVED		0x000F-0x00FF		
Vendor Specific		0x0100-0x01FF		
RESERVED		0x0200-0x7FFF		

The following are iSNSP response messages used in support of iSCSI:

Response Message Desc	Abbreviation	Func_ID	REQUIRED for: Server	Client
RESERVED		0x8000		
Device Attr Register Rsp	DevAttrRegRsp	0x8001	*	*
Device Attr Query Rsp	DevAttrQryRsp	0x8002	*	*
Device Get Next Rsp	DevGetNextRsp	0x8003	*	
Device Dereg Rsp	DevDeregRsp	0x8004	*	*
SCN Register Rsp	SCNRegRsp	0x8005	*	

SCN Deregister Rsp	SCNDeregRsp	0x8006	*	
SCN Event Rsp	SCNEventRsp	0x8007	*	
SCN Response	SCNRsp	0x8008	*	
DD Register Rsp	DDRegRsp	0x8009	*	*
DD Deregister Rsp	DDDeregRsp	0x800A	*	*
DDS Register Rsp	DDSRegRsp	0x800B	*	*
DDS Deregister Rsp	DDSDeregRsp	0x800C	*	*
Entity Stat Inquiry Rsp	ESIRsp	0x800D	*	
RESERVED		0x800E-0x80FF		
Vendor Specific		0x8100-0x81FF		
RESERVED		0x8200-0xFFFF		

4.2. iFCP Requirements

In iFCP, use of iSNS is REQUIRED. No alternatives exist for support of iFCP Naming & Discovery functions.

4.2.1. Required Attributes for Support of iFCP

The following table displays attributes that are used by iSNS to support iFCP. Attributes indicated in the REQUIRED for Server column MUST be implemented by the iSNS server that supports iFCP. Attributes indicated in the REQUIRED for Client column MUST be supported by iFCP gateways. Attributes indicated in the K (Key) column uniquely identify the object type in the iSNS Server. A more detailed description of each attribute is found in Section 6.

			REQUIRED for:	
Object	Attribute	K	Server	Client
NETWORK ENTITY	Entity Identifier	*	*	*
	Entity Protocol		*	*
	Management IP Address		*	
	Timestamp		*	
	Protocol Version Range		*	
	Registration period			
	Entity Index			
	Entity IKE Phase-1 Proposal			
	Entity Certificate			
PORTAL	IP Address	*	*	*
	TCP/UDP Port	*	*	*
	Symbolic Name		*	
	ESI Interval		*	
	ESI Port		*	
	SCN Port		*	
	Portal IKE Phase-1 Proposal			
	Portal IKE Phase-2 Proposal			

	Portal Certificate			
	Security Bitmap		*	
STORAGE NODE (FC Port)	FC Port Name (WWPN)	*	*	*
	Port_ID		*	*
	FC Port Type		*	*
	Port Symbolic Name		*	
	Fabric Port Name (FWWN)		*	
	Hard Address		*	
	Port IP Address		*	
	Class of Service		*	
	FC FC-4 Types		*	
	FC FC-4 Descriptors		*	
	FC FC-4 Features		*	
	SCN Bitmap		*	
	iFCP Port Role		*	
	Permanent Port Name		*	
FC DEVICE (FC Node)	FC Node Name (WWNN)	*	*	*
	Node Symbolic Name		*	
	Node IP Address		*	
	Node IPA		*	
	Proxy iSCSI Name			
DISCOVERY DOMAIN	DD ID	*	*	*
	DD Symbolic Name		*	
	DD Member FC Port Name		*	
	DD Member Portal Index		*	
	DD Member Portal IP Addr		*	
	DD Member Portal TCP/UDP		*	
DISCOVERY DOMAIN SET	DDS ID	*	*	
	DDS Symbolic Name		*	
	DDS Status		*	
OTHER	Switch Name			
	Preferred_ID			
	Assigned_ID			
	Virtual_Fabric_ID			

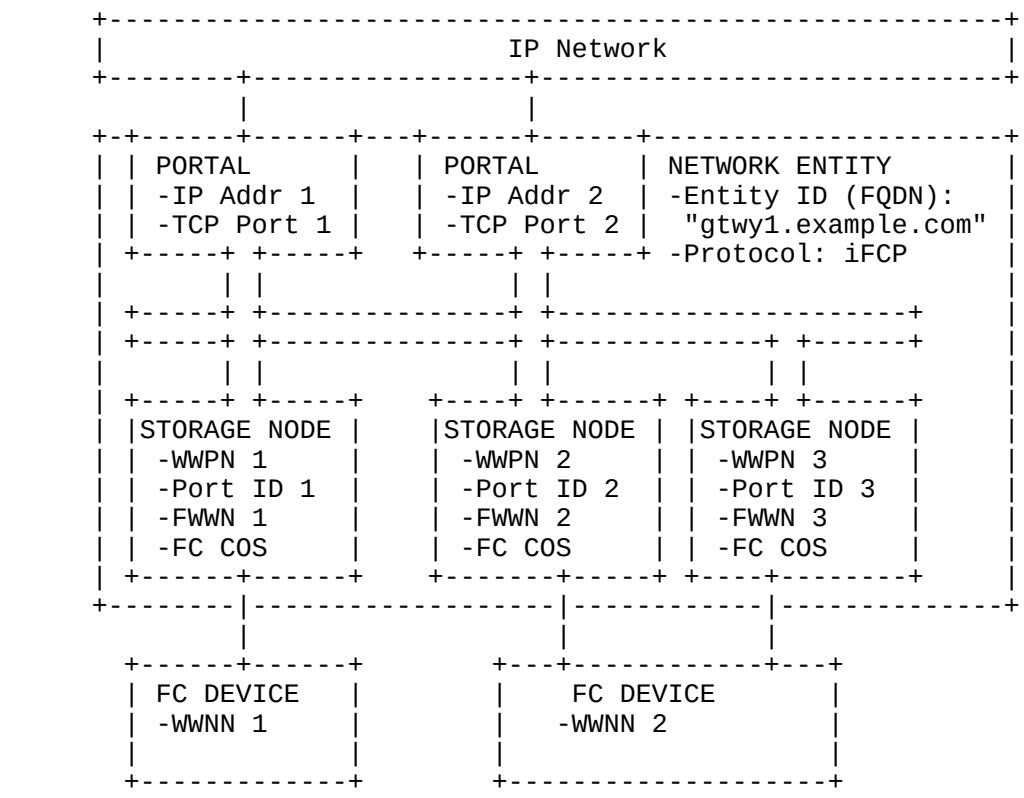
All iFCP user-specified and vendor-specified attributes are OPTIONAL to implement and use.

4.2.2. Example: iFCP Object Model Diagram

The iFCP protocol allows native Fibre Channel devices or Fibre Channel fabrics connected to an iFCP gateway to be directly internetworked using IP.

When supporting iFCP, the iSNS server stores Fibre Channel device attributes, iFCP gateway attributes, and Fibre Channel fabric switch attributes that might also be stored in an FC name server.

The following diagram shows a representation of a gateway supporting multiple Fibre Channel devices behind it. The two Portal objects represent IP interfaces on the iFCP gateway that can be used to access any of the three Storage Node objects behind it. Note that the FC Device object is not contained in the Network Entity object. However, each FC Device has a relationship to one or more Storage Node objects.



4.2.3. Required Commands and Response Messages for Support of iFCP

The iSNSP messages and responses displayed in the following tables are available to support iFCP gateways. Messages indicated in the REQUIRED TO IMPLEMENT column MUST be supported by the iSNS server used by iFCP gateways. Messages indicated in the REQUIRED TO USE column MUST be supported by the iFCP gateways themselves.

Message Description	Abbreviation	Func ID	REQUIRED for: Server	Client
RESERVED		0x0000		
Device Attr Reg Request	DevAttrReg	0x0001	*	*
Device Attr Query Request	DevAttrQry	0x0002	*	*
Device Get Next Request	DevGetNext	0x0003	*	
Device Dereg Request	DevDereg	0x0004	*	*
SCN Register Request	SCNReg	0x0005	*	
SCN Deregister Request	SCNDereg	0x0006	*	
SCN Event	SCNEvent	0x0007	*	
State Change Notification	SCN	0x0008	*	
DD Register	DDReg	0x0009	*	*
DD Deregister	DDDereg	0x000A	*	*
DDS Register	DDSReg	0x000B	*	*
DDS Deregister	DDSDereg	0x000C	*	*
Entity Status Inquiry	ESI	0x000D	*	
Name Service Heartbeat	Heartbeat	0x000E	*	
Reserved	Reserved	0x000F-0x0010		
Request FC_DOMAIN_ID	RqstDomId	0x0011		
Release FC_DOMAIN_ID	RlseDomId	0x0012		
Get FC_DOMAIN_IDs	GetDomId	0x0013		
RESERVED		0x0014-0x00FF		
Vendor Specific		0x0100-0x01FF		
RESERVED		0x0200-0x7FFF		

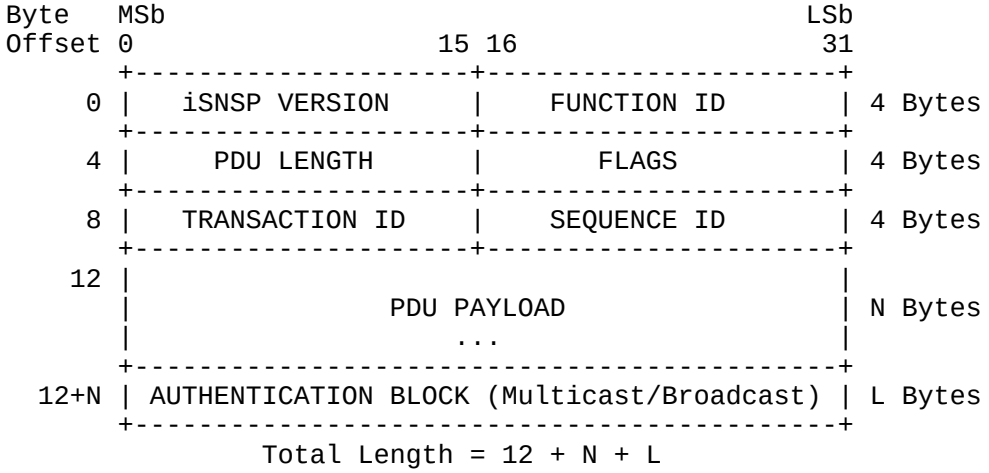
The following are iSNSP response messages in support of iFCP:

Response Message Desc	Abbreviation	Func_ID	REQUIRED for: Server	Client
RESERVED		0x8000		
Device Attr Reg Rsp	DevAttrRegRsp	0x8001	*	*
Device Attr Query Rsp	DevAttrQryRsp	0x8002	*	*
Device Get Next Rsp	DevGetNextRsp	0x8003	*	
Device Deregister Rsp	DevDeregRsp	0x8004	*	*
SCN Register Rsp	SCNRegRsp	0x8005	*	
SCN Deregister Rsp	SCNDeregRsp	0x8006	*	
SCN Event Rsp	SCNEventRsp	0x8007	*	
SCN Rsp	SCNRsp	0x8008	*	

DD Register Rsp	DDRegRsp	0x8009	*	*
DD Deregister Rsp	DDDeregRsp	0x800A	*	*
DDS Register Rsp	DDSRegRsp	0x800B	*	*
DDS Deregister Rsp	DDSDeregRsp	0x800C	*	*
Entity Status Inquiry Rsp	ESIRsp	0x800D	*	
NOT USED		0x800E		
RESERVED		0x800F-0x8010		
Request FC_DOMAIN_ID Rsp	RqstDomIdRsp	0x8011		
Release FC_DOMAIN_ID Rsp	RlseDomIdRsp	0x8012		
Get FC_DOMAIN_IDs	GetDomIdRsp	0x0013		
RESERVED		0x8014-0x80FF		
Vendor Specific		0x8100-0x81FF		
RESERVED		0x8200-0xFFFF		

5. iSNSP Message Format

The iSNSP message format is similar to the format of other common protocols such as DHCP, DNS and BOOTP. An iSNSP message may be sent in one or more iSNS Protocol Data Units (PDU). Each PDU is 4-byte aligned. The following describes the format of the iSNSP PDU:



5.1. iSNSP PDU Header

The iSNSP PDU header contains the iSNSP VERSION, FUNCTION ID, PDU LENGTH, FLAGS, TRANSACTION ID, and SEQUENCE ID fields as defined below.

5.1.1. iSNSP Version

The iSNSP version described in this document is 0x0001. All other values are RESERVED. The iSNS server MAY reject messages for iSNSP version numbers that it does not support.

5.1.2. iSNSP Function ID

The FUNCTION ID defines the type of iSNS message and the operation to be executed. FUNCTION_ID values with the leading bit cleared indicate query, registration, and notification messages, whereas FUNCTION_ID values with the leading bit set indicate response messages.

See Section 4 under the appropriate protocol (i.e., iSCSI or iFCP) for a mapping of the FUNCTION_ID value to the iSNSP Command or Response message. All PDUs comprising an iSNSP message must have the same FUNCTION_ID value.

5.1.3. iSNSP PDU Length

The iSNS PDU Length specifies the length of the PDU PAYLOAD field in bytes. The PDU Payload contains TLV attributes for the operation.

Additionally, response messages contain a success/failure code. The PDU Length MUST be 4-byte aligned.

5.1.4. iSNSP Flags

The FLAGS field indicates additional information about the message and the type of Network Entity that generated the message. The following table displays the valid flags:

Bit Position	Enabled (1) means:
16	Sender is the iSNS client
17	Sender is the iSNS server
18	Authentication block is present
19	Replace flag (for DevAttrReg)
20	Last PDU of the iSNS message
21	First PDU of the iSNS message
22-31	RESERVED

5.1.5. iSNSP Transaction ID

The TRANSACTION ID MUST be set to a unique value for each concurrently outstanding request message. Replies MUST use the same TRANSACTION ID value as the associated iSNS request message. If a

message is retransmitted, the original TRANSACTION ID value MUST be used. All PDUs comprising an iSNSP message must have the same TRANSACTION ID value.

5.1.6. iSNSP Sequence ID

The SEQUENCE ID has a unique value for each PDU within a single transaction. The SEQUENCE_ID value of the first PDU transmitted in a given iSNS message MUST be zero (0), and each SEQUENCE_ID value in each PDU MUST be numbered sequentially in the order in which the PDUs are transmitted. Note that the two-byte SEQUENCE ID allows for up to 65536 PDUs per iSNS message.

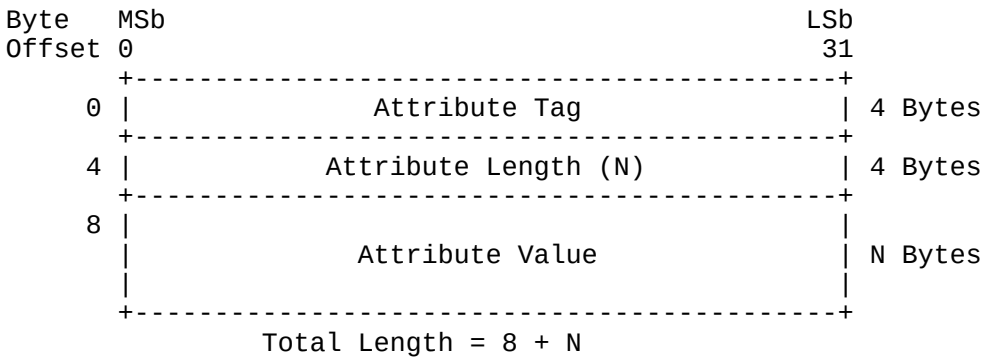
5.2. iSNSP Message Segmentation and Reassembly

iSNS messages may be carried in one or more iSNS PDUs. If only one iSNS PDU is used to carry the iSNS message, then bit 21 (First PDU) and bit 20 in the FLAGS field (Last PDU) SHALL both be set. If multiple PDUs are used to carry the iSNS message, then bit 21 SHALL be set in the first PDU of the message, and bit 20 SHALL be set in the last PDU.

All PDUs comprising the same iSNSP message SHALL have the same FUNCTION_ID and TRANSACTION_ID values. Each PDU comprising an iSNSP message SHALL have a unique SEQUENCE_ID value.

5.3. iSNSP PDU Payload

The iSNSP PDU PAYLOAD is of variable length and contains attributes used for registration and query operations. The attribute data items use a format similar to that of other protocols, such as DHCP [RFC2131] options. Each iSNS attribute is specified in the PDU Payload using Tag-Length-Value (TLV) data format, as shown below:



Attribute Tag: a 4-byte field that identifies the attribute as defined in Section 6.1. This field contains the tag value from the indicated table.

Attribute Length: a 4-byte field that indicates the length, in bytes, of the value field to follow in the TLV. For variable-length attributes, the value field MUST contain padding bytes, if necessary, in order to achieve 4-byte alignment. A "zero-length TLV" contains only the attribute tag and length fields.

Attribute Value: a variable-length field containing the attribute value and padding bytes (if necessary).

The above format is used to identify each attribute in the PDU Payload. Note that TLV boundaries need not be aligned with PDU boundaries; PDUs may carry one or more TLVs, or any fraction thereof. The Response Status Code, contained in response message PDU Payloads and described below, is not in TLV format. PDU Payloads for messages that do not contain iSNS attributes, such as the Name Service Heartbeat, do not use the TLV format.

5.3.1. Attribute Value 4-Byte Alignment

All attribute values are aligned to 4-byte boundaries. For variable length attributes, if necessary, the TLV length MUST be increased to the next 4-byte boundary through padding with bytes containing zero (0). If an attribute value is padded, a combination of the tag and attribute value itself is used to determine the actual value length and number of pad bytes. There is no explicit count of the number of pad bytes provided in the TLV.

5.4. iSNSP Response Status Codes

All iSNSP response messages contain a 4-byte Status Code field as the first field in the iSNSP PDU PAYLOAD. If the original iSNSP request message was processed normally by the iSNS server, or by the iSNS client for ESI and SCN messages, then this field SHALL contain a status code of 0 (Successful). A non-zero status code indicates rejection of the entire iSNS client request message.

Status Code	Status Description
0	Successful
1	Unknown Error
2	Message Format Error
3	Invalid Registration
4	RESERVED
5	Invalid Query
6	Source Unknown
7	Source Absent
8	Source Unauthorized
9	No Such Entry
10	Version Not Supported
11	Internal Error
12	Busy
13	Option Not Understood
14	Invalid Update
15	Message (FUNCTION_ID) Not Supported
16	SCN Event Rejected
17	SCN Registration Rejected
18	Attribute Not Implemented
19	FC_DOMAIN_ID Not Available
20	FC_DOMAIN_ID Not Allocated
21	ESI Not Available
22	Invalid Deregistration
23	Registration Feature Not Supported
24 and above	RESERVED

5.5. Authentication for iSNS Multicast and Broadcast Messages

For iSNS multicast and broadcast messages (see Section 2.9.3), the iSNSP provides authentication capability. The following section details the iSNS Authentication Block, which is identical in format to the SLP authentication block [RFC2608]. iSNS unicast messages SHOULD NOT include the authentication block, but rather should rely upon IPSec security mechanisms.

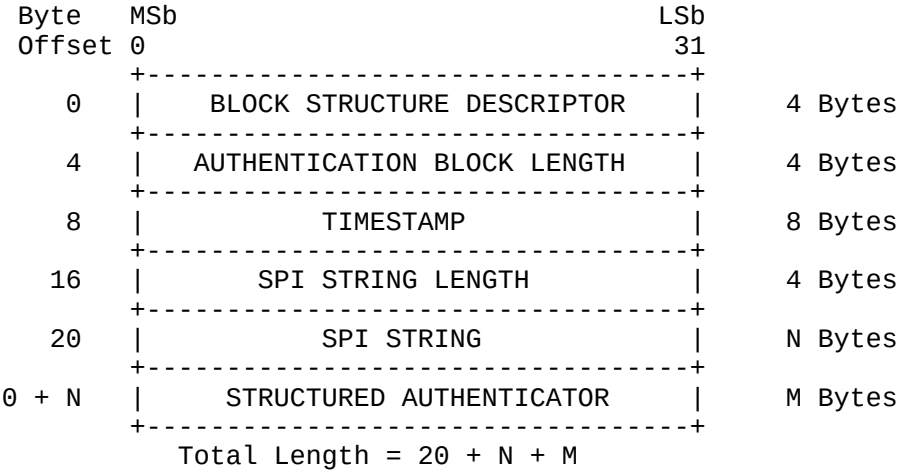
If a message contains an authentication block, then the "Authentication block present" bit in the iSNSP PDU header FLAGS field SHALL be enabled.

If a PKI is available with an [X.509] Certificate Authority (CA), then public key authentication of the iSNS server is possible. The authentication block leverages the DSA with SHA-1 algorithm, which can easily integrate into a public key infrastructure.

The authentication block contains a digital signature for the multicast message. The digital signature is calculated on a per-PDU basis. The authentication block contains the following information:

1. A time stamp, to prevent replay attacks.
2. A structured authenticator containing a signature calculated over the time stamp and the message being secured.
3. An indicator of the cryptographic algorithm that was used to calculate the signature.
4. An indicator of the keying material and algorithm parameters, used to calculate the signature.

The authentication block is described in the following figure:



BLOCK STRUCTURE DESCRIPTOR (BSD): Defines the structure and algorithm to use for the STRUCTURED AUTHENTICATOR. BSD values from 0x00000000 to 0x00007FFF are assigned by IANA, while values 0x00008000 to 0x00008FFF are for private use.

AUTHENTICATION BLOCK LENGTH: Defines the length of the authentication block, beginning with the BSD field and running through the last byte of the STRUCTURED AUTHENTICATOR.

TIMESTAMP: This is an 8-byte unsigned, fixed-point integer giving the number of seconds since 00:00:00 GMT on January 1, 1970.

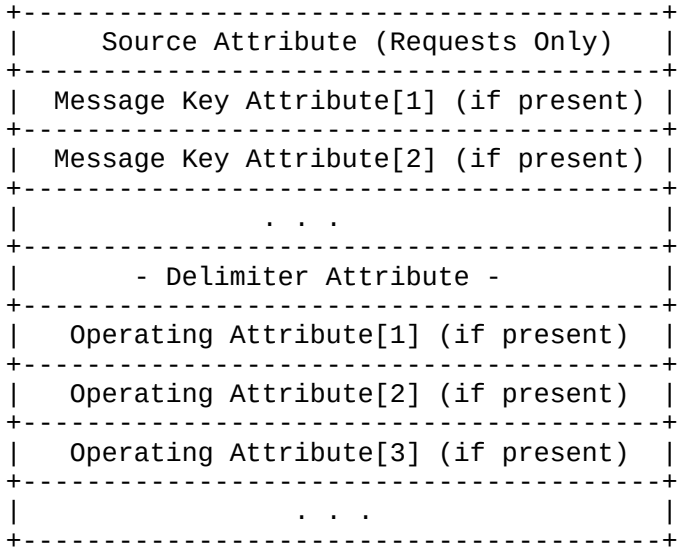
SPI STRING LENGTH: The length of the SPI STRING field.

SPI STRING (Security Parameters Index): Index to the key and algorithm used by the message recipient to decode the STRUCTURED AUTHENTICATOR field.

STRUCTURED AUTHENTICATOR: Contains the digital signature. For the default BSD value of 0x0002, this field SHALL contain the binary ASN.1 encoding of output values from the DSA with SHA-1 signature calculation as specified in Section 2.2.2 of [RFC3279].

5.6. Registration and Query Messages

The iSNSP registration and query message PDU Payloads contain a list of attributes, and have the following format:



Each Source, Message Key, Delimiter, and Operating attribute is specified in the PDU Payload using the Tag-Length-Value (TLV) data format. iSNS Registration and Query messages are sent by iSNS Clients

to the iSNS server IP Address and well-known TCP/UDP Port. The iSNS Responses will be sent to the iSNS Client IP address and TCP/UDP port number from the original request message.

5.6.1. Source Attribute

The Source Attribute is used to identify the Storage Node to the iSNS server for queries and other messages that require source identification. The Source Attribute uniquely identifies the source of the message. Valid Source Attribute types are shown below.

Valid Source Attributes
iSCSI Name
FC Port Name WWPN

For a query operation, the Source Attribute is used to limit the scope of the specified operation to the Discovery Domains of which the source is a member. Special Control Nodes, identified by the Source Attribute, may be administratively configured to perform the specified operation on all objects in the iSNS database without scoping to Discovery Domains.

For messages that change the contents of the iSNS database, the iSNS server MUST verify that the Source Attribute identifies either a Control Node or a Storage Node that is a part of the Network Entity containing the added, deleted, or modified objects.

5.6.2. Message Key Attributes

Message Key attributes are used to identify matching objects in the iSNS database for iSNS query and registration messages. If present, the Message Key MUST be a Registration or Query Key for an object as described in Sections 5.6.5 and 6.1. A Message Key is not required when a query spans the entire set of objects available to the Source or a registration is for a new Entity.

iSCSI Names used in the Message Key MUST be normalized according to the stringprep template [STRINGPREP]. Entity Identifiers (EIDs) used in the Message Key MUST be normalized according to the nameprep template [NAMEPREP].

5.6.3. Delimiter Attribute

The Delimiter Attribute separates the Message Key attributes from the Operating Attributes in a PDU Payload. The Delimiter Attribute has a tag value of 0 and a length value of 0. The Delimiter Attribute is always 8 bytes long (a 4-byte tag field and a 4-byte length field,

all containing zeros). If a Message Key is not required for a message, then the Delimiter Attribute immediately follows the Source Attribute.

5.6.4. Operating Attributes

The Operating Attributes are a list of one or more key and non-key attributes related to the actual iSNS registration or query operation being performed.

Operating Attributes include object key attributes and non-key attributes. Object key attributes uniquely identify iSNS objects. Key attributes MUST precede the non-key attributes of each object in the Operating Attributes. The tag value distinguishes the attribute as an object key attribute (i.e., tag=1, 16&17, 32, 64, and 96) or a non-key attribute. iSCSI Names used in the Operating Attributes MUST be normalized according to the stringprep template [STRINGPREP]. Entity Identifiers (EIDs) used in the Operating Attributes MUST be normalized according to the nameprep template [NAMEPREP].

The ordering of Operating Attributes in the message is important for determining the relationships among objects and their ownership of non-key attributes. iSNS protocol messages that violate these ordering rules SHALL be rejected with the Status Code of 2 (Message Format Error). See the message descriptions for proper operating attribute ordering requirements.

Some objects are keyed by more than one object key attribute value. For example, the Portal object is keyed by attribute tags 16 and 17. When describing an object keyed by more than one key attribute, every object key attribute of that object MUST be listed sequentially by tag value in the message before non-key attributes of that object and key attributes of the next object. A group of key attributes of this kind is treated as a single logical key attribute when identifying an object.

Non-key attributes that immediately follow key attributes MUST be attributes of the object referenced by the key attributes. All non-key attributes of an object MUST be listed before the object key attributes introducing the next object.

Objects MUST be listed in inheritance order, according to their containment order. Storage Node and Portal objects and their respective attributes MUST follow the Network Entity object to which they have a relationship. Similarly, FC Device objects MUST follow the Storage Node object to which they have a relationship.

Vendor-specific objects defined by tag values in the range 1537-2048 have the same requirements described above.

5.6.4.1. Operating Attributes for Query and Get Next Requests

In Query and Get Next request messages, TLV attributes with length value of 0 are used to indicate which Operating Attributes are to be returned in the corresponding response. Operating Attribute values that match the TLV attributes in the original message are returned in the response message.

5.6.5. Registration and Query Request Message Types

The following describes each query and message type.

5.6.5.1. Device Attribute Registration Request (DevAttrReg)

The DevAttrReg message type is 0x0001. The DevAttrReg message provides the means for iSNS clients to update existing objects or register new objects. The value of the replace bit in the FLAGs field determines whether the DevAttrReg message updates or replaces an existing registration.

The Source Attribute identifies the Node initiating the registration request.

The Message Key identifies the object the DevAttrReg message acts upon. It MUST contain the key attribute(s) identifying an object. This object MUST contain all attributes and related subordinate object attributes that will be included in the Operating Attributes of the DevAttrReg PDU Payload. The key attribute(s) identifying this object MUST also be included among the Operating Attributes.

If the Message Key contains an EID and no pre-existing objects match the Message Key, then the DevAttrReg message SHALL create a new Entity with the specified EID and any new object(s) specified by the Operating Attributes. The replace bit SHALL be ignored.

If the Message Key does not contain an EID, and no pre-existing objects match the Message Key, then the DevAttrReg message SHALL be rejected with a status code of 3 (Invalid Registration).

If the Message Key is not present, then the DevAttrReg message implicitly registers a new Network Entity. In this case, the replace bit SHALL be ignored; a new Network Entity SHALL be created. Existing entities, their objects, and their relationships remain unchanged.

The replace bit determines the kind of operation conducted on the object identified in the DevAttrReg Message Key. The replace bit only applies to the DevAttrReg message; it is ignored for all other message types.

If the replace bit is set, then the objects, attributes, and relationships specified in the Operating Attributes SHALL replace the object identified by the Message Key. The object and all of its subordinate objects SHALL be deregistered, and the appropriate SCNs SHALL be sent by the iSNS server for the deregistered objects. The objects listed in the Operating Attributes are then used to replace the just-deregistered objects. Note that additional SCNs SHALL be sent for the newly-registered objects, if appropriate. Existing objects and relationships that are not identified or that are subordinate to the object identified by the Message Key MUST NOT be affected or changed.

If the replace bit is not set, then the message updates the attributes of the object identified by the Message Key and its subordinate objects. Existing object containment relationships MUST NOT be changed. For existing objects, key attributes MUST NOT be modified, but new subordinate objects MAY be added.

The Operating Attributes represent objects, attributes, and relationships that are to be registered. Multiple related objects and attributes MAY be registered in a single DevAttrReg message. The ordering of the objects in this message indicates the structure of, and associations among, the objects to be registered. At least one object MUST be listed in the Operating Attributes. Additional objects (if any) MUST be subordinate to the first object listed. Key attributes MUST precede non-key attributes of each object. A given object may only appear a maximum of once in the Operating Attributes of a message. If the Node identified by the Source Attribute is not a Control Node, then the objects in the operating attributes MUST be members of the same Network Entity as the Source Node.

For example, to establish relationships between a Network Entity object and its Portal and Storage Node objects, the Operating Attributes list the key and non-key attributes of the Network Entity object, followed by the key and non-key attributes of each Portal and Storage Node object to be linked to that Network Entity. Similarly, an FC Device object that follows a Storage Node object is considered subordinate to that Storage Node.

New PG objects are registered when an associated Portal or iSCSI Node object is registered. An explicit PG object registration MAY follow a Portal or iSCSI Node object registration in a DevAttrReg message.

When a Portal is registered, the Portal attributes MAY immediately be followed by a PGT attribute. The PGT attribute SHALL be followed by the set of PG iSCSI Names representing nodes that will be associated to the Portal using the indicated PGT value. Additional sets of PGTs and PG iSCSI Names to be associated to the registered Portal MAY follow. Indicated PGT values are assigned to the PG object associated with the newly registered Portal and to the iSCSI Storage Node(s) referenced immediately following the PGT attribute in the operating attributes.

When an iSCSI Storage Node is registered, the Storage Node attributes MAY immediately be followed by a PGT attribute. The PGT attribute SHALL be followed by the set of PG Portal IP-Address, PG TCP/UDP Port pairs representing Portal objects that will be associated with the Storage Node using the indicated PGT value. Additional sets of PGTs and PG Portal IP-Address PG TCP/UDP Port pairs to be associated with the registered Storage Node MAY follow. Indicated PGT values are assigned to the PG object associated with the newly registered iSCSI Storage Node and Portal object(s) referenced immediately following the PGT attribute in the operating attributes.

If the PGT value is not included in the Storage Node or Portal object registration, and if a PGT value was not previously registered for the relationship, then the PGT for the corresponding PG object SHALL be registered with a value of 0x00000001. If the PGT attribute is included in the registration message as a 0-length TLV, then the PGT value for the corresponding PG object SHALL be registered as NULL. A 0-length TLV for the PGT in an update registration message overwrites the previous PGT value with NULL, indicating that there is no relationship between the Storage Node and Portal.

A maximum of one Network Entity object can be created or updated with a single DevAttrReg message. Consequently, the Operating Attributes MUST NOT contain more than one Network Entity object. There is no limit to the number of Portal, Storage Node, and FC Device objects that can listed in the Operating Attributes, provided they are all subordinate to the listed Network Entity object.

If the Message Key and Operating Attributes do not contain an EID attribute, or if the EID attribute has a length of 0, then a new Network Entity object SHALL be created and the iSNS server SHALL supply a unique EID value for it. The assigned EID value SHALL be included in the DevAttrReg Response message. If the Message Key and Operating Attributes contain an EID that does not match the EID of an existing Network Entity in the iSNS database, then a new Network Entity SHALL be created and assigned the value contained in that EID attribute. Finally, if the Message Key and Operating Attributes contain an EID that matches the EID of an existing object in the iSNS

database, then the objects, attributes, and relationships specified in the Operating Attributes SHALL be appended to the existing Network Entity identified by the EID.

A registration message that creates a new Network Entity object MUST contain at least one Portal or one Storage Node. If the message does not, then it SHALL be considered invalid and result in a response with Status Code of 3 (Invalid Registration).

If an iSNS Server does not support a registration feature, such as explicit PG object registration, then the server SHALL return a Status Code of 23 (Registration Feature Not Supported).

Note that the iSNS server may modify or reject the registration of certain attributes, such as ESI Interval. In addition, the iSNS server may assign values for additional Operating Attributes that are not explicitly registered in the original DevAttrReg message, such as the EID and WWNN Token.

5.6.5.2. Device Attribute Query Request (DevAttrQry)

The DevAttrQry message type is 0x0002. The DevAttrQry message provides an iSNS client with the means to query the iSNS server for object attributes.

The Source Attribute identifies the Node initiating the request. For non-Control Nodes initiating the DevAttrQry message, the query is scoped to the Discovery Domains of which the initiating Node is a member. The DevAttrQry message SHALL only return information on Storage Nodes and their related parent and subordinate objects, where the Storage Node has a common Discovery Domain with the Node identified in the Source Attribute.

The Message Key may contain key or non-key attributes or no attributes at all. If multiple attributes are used as the Message Key, then they MUST all be from the same object type (e.g., IP address and TCP/UDP Port are attributes of the Portal object type). A Message Key with non-key attributes may match multiple instances of the specific object type. A Message Key with zero-length TLV(s) is scoped to every object of the type indicated by the zero-length TLV(s). An empty Message Key field indicates the query is scoped to the entire database accessible by the source Node.

The DevAttrQry response message returns attributes of objects listed in the Operating Attributes that are related to the Message Key of the original DevAttrQry message. The Operating Attributes of the DevAttrQry message contain zero-length TLVs that specify the attributes that are to be returned in the DevAttrQryRsp message. A

Message Key containing zero-length TLVs indicates that the set of attributes specified in the Operating Attributes are to be returned for each object matching the type indicated by the Message Key.

If the Message Key contains non-zero length TLVs, then Operating Attributes for the object matching the Message Key SHALL be returned in the DevAttrQryRsp message. Each attribute type (i.e., zero-length TLV) in the Operating Attributes indicates an attribute from the object matching the Message Key, or from other objects in the same Entity having a relationship to the object matching the Message Key, is to be returned in the response. The ordering of the object keys and associated attributes returned in the DevAttrQry response message SHALL be the same as in the original query message. If no objects match the Message Key, then the DevAttrQryRsp message SHALL NOT return any operating attributes. Such a message and its corresponding response SHALL NOT be considered an error.

The Portal Group object determines whether a relationship exists between a given Storage Node and Portal object. If the PGT of the Portal Group is not NULL, then a relationship exists between the indicated Storage Node and Portal; if the PGT is NULL, then no relationship exists. Therefore, the value (NULL or not NULL) of the PGT attribute of each Portal Group object determines the structure and ordering of the DevAttrQry response to a query for Storage Nodes and Portals.

For example, an iSNS database contains a Network Entity having two Portals and two Nodes. Each Storage Node has two Portal Groups, one with a NULL PGT value for one Portal and another with a non-NULL PGT value for the other Portal. The DevAttrQry message contains a Message Key entry matching one of the Nodes, and Operating Attributes with zero-length TLVs listing first the Node attributes, Portal attributes, and then the PG attributes. The response message SHALL therefore return first the matching Node object, then the requested attributes of the one Portal object that can be used to access the Storage Node (as indicated by the PGT), and finally the requested attributes of the PG object used to access that Storage Node. The order in which each object's attributes are listed is the same as the ordering of the object's attributes in the Operating Attributes of the original request message.

If the Message Key Attribute contains zero-length TLV(s), then the query returns requested attributes for all objects matching the Message Key type (DD restrictions SHALL apply for non-Control Nodes). If multiple objects match the Message Key type, then the attributes for each object matching the Message Key MUST be listed before the attributes for the next matching object are listed in the query

response. In other words, the process described above must be iterated in the message response for each object that matches the Message Key type specified by the zero-length TLV(s).

For example, an iSNS database contains only one Network Entity having two Portals and three Nodes. All PG objects in the Entity have a PGT value of 0x00000001. In the DevAttrQry message, the Message Key contains a zero-length TLV specifying a Node type, and Operating Attributes listing first the Node attributes, and then the Portal attributes. The response message will return, in the following order, the attributes for the first, next, and last Node objects, each followed by attributes for both Portals. If that same DevAttrQry message had instead contained a zero-length TLV specifying the Network Entity type, then the response message would have returned attributes for all three Node objects, followed by attributes for the two Portals.

If there is no Message Key Attribute, then the query returns all attributes in the iSNS database (once again, DD restrictions SHALL apply for non-Control Nodes). All attributes matching the type specified by each zero-length TLV in the Operating Attributes SHALL be listed. All attributes of each type SHALL be listed before the attributes matching the next zero-length TLV are listed.

For example, an iSNS database contains two Entities, each having two Nodes and two Portals. The DevAttrQry message contains no Message Key attribute, and Operating Attributes list first the Portal attributes, and then the Node attributes. The Operating Attributes of the response message will return attributes from each of the four Portals, followed by attributes from each of the four nodes.

If a DevAttrQry message requests an attribute for which the iSNS server has no value, then the server SHALL NOT return the requested attribute in the query response. Such query and response messages SHALL NOT be considered errors.

Registration and query messages for iSNS server-specific attributes (i.e., tags in the range 132 to 384) SHALL be formatted using the identifying key attribute of the Storage Node originating the query (i.e., iSCSI Name or FC Port Name WWPN) for both the Source Attribute and Message Key attribute. Operating Attributes SHALL include the TLV of the server-specific attribute being requested.

DD membership can be discovered through the DevAttrQry message by including either DD member attributes (i.e., DD Member iSCSI Index, DD Member iSCSI Node, DD Member iFCP Node, DD Member Portal Index, DD Member Portal IP Addr, and DD Member Portal TCP/UDP) or the object key of the Storage Node or Portal (i.e., iSCSI Name, iSCSI Index,

Portal IP Addr, Portal TCP/UDP Port, and Portal Index) in the Operating Attributes. Using DD member attributes SHALL return both registered and unregistered member Storage Nodes and/or Portals of a DD. DevAttrQry messages using the Storage Node and/or Portal object key SHALL return only member Storage Nodes or Portals that are currently registered in the iSNS database.

The DevAttrQry message SHALL support the following minimum set of Message Key Attributes:

Valid Message Key Attributes for Queries
Entity Identifier
Entity Protocol
Portal IP-Address & Portal TCP/UDP Port
Portal Index
iSCSI Node Type
iSCSI Name
iSCSI Index
PG Index
FC Port Name WWPN
FC Port Type
FC-4 Type
Discovery Domain ID
Discovery Domain Set ID
Source Attribute (for server-specific attributes)
Switch Name (FC Device WWNN--for Virtual_Fabric_ID queries)

5.6.5.3. Device Get Next Request (DevGetNext)

The DevGetNext message type is 0x0003. This message provides the iSNS client with the means to retrieve each and every instance of an object type exactly once.

The Source Attribute identifies the Node initiating the DevGetNext request, and is used to scope the retrieval process to the Discovery Domains of which the initiating Node is a member.

The Message Key Attribute may be an Entity Identifier (EID), iSCSI Name, iSCSI Index, Portal IP Address and TCP/UDP Port, Portal Index, PG Index, FC Node Name WWNN, or FC Port Name WWPN. If the TLV length of the Message Key Attribute(s) is zero, then the first object entry in the iSNS database matching the Message Key type SHALL be returned in the Message Key of the corresponding DevGetNextRsp message. If non-zero-length TLV attributes are contained in the Message Key, then the DevGetNext response message SHALL return the next object stored after the object identified by the Message Key in the original DevGetNext request message.

If the Message Key provided matches the last object instance in the iSNS database, then the Status Code of 9 (No Such Entry) SHALL be returned in the response.

The Operating Attributes can be used to specify the scope of the DevGetNext request, and to specify the attributes of the next object, which are to be returned in the DevGetNext response message. All Operating Attributes MUST be attributes of the object type identified by the Message Key. For example, if the Message Key is an Entity_ID attribute, then the Operating Attributes MUST NOT contain attributes of Portals.

Non-zero-length TLV attributes in the Operating Attributes are used to scope the DevGetNext message. Only the next object with attribute values that match the non-zero-length TLV attributes SHALL be returned in the DevGetNext response message.

Zero-length TLV attributes MUST be listed after non-zero-length attributes in the Operating Attributes of the DevGetNext request message. Zero-length TLV attributes specify the attributes of the next object which are to be returned in the DevGetNext response message.

Note that there are no specific requirements concerning the order in which object entries are retrieved from the iSNS database; the retrieval order of object entries using the DevGetNext message is implementation specific.

The iSNS client is responsible for ensuring that information acquired through use of the DevGetNext message is accurate and up-to-date. There is no assurance that the iSNS database will not change between successive DevGetNext request messages. If the Message Key provided does not match an existing database entry, then attributes for the next object key following the provided Message Key SHALL be returned. For example, an object entry may have been deleted between successive DevGetNext messages. This may result in a DevGetNext request in which the Message Key does not match an existing object entry. In this case, attributes for the next object stored in the iSNS database are returned.

5.6.5.4. Device Deregister Request (DevDereg)

The DevDereg message type is 0x0004. This message is used to remove object entries from the iSNS database. One or more objects may be removed through a single DevDereg message. Note that deregistered Storage Node objects will retain membership in their Discovery Domain(s) until explicit deregistration of the membership(s) or Discovery Domain(s).

Upon receiving the DevDereg, the iSNS server removes all objects identified by the Operating Attribute(s), and all subordinate objects that are solely dependent on those identified objects. For example, removal of a Network Entity also results in removal of all associated Portal, Portal Group, Storage Node, and FC Device objects associated with that Network Entity. FC Device objects SHALL not be deregistered in this manner unless all Storage Nodes associated with them have been deregistered.

The DevDereg request PDU Payload contains a Source Attribute and Operating Attribute(s); there are no Message Key Attributes. If the Node identified by the Source Attribute is not a Control Node, then it MUST be from the same Network Entity as the object(s) identified for removal by the Operating Attribute(s). Valid Operating Attributes are shown below:

Valid Operating Attributes for DevDereg
Entity Identifier
Portal IP-Address & Portal TCP/UDP Port
Portal Index
iSCSI Name
iSCSI Index
FC Port Name WWPN
FC Node Name WWNN

The removal of the object may result in SCN messages to the appropriate iSNS clients.

Attempted deregistration of non-existing entries SHALL not be considered an error.

If all Nodes and Portals associated with a Network Entity are deregistered, then the Network Entity SHALL also be removed.

If both the Portal and iSCSI Storage Node objects associated with a Portal Group object are removed, then that Portal Group object SHALL also be removed. The Portal Group object SHALL remain registered as long as either of its associated Portal or iSCSI Storage Node objects remain registered. If a deleted Storage Node or Portal object is subsequently re-registered, then a relationship between the re-registered object and an existing Portal or Storage Node object registration, indicated by the PG object, SHALL be restored.

5.6.5.5. SCN Register Request (SCNReg)

The SCNReg message type is 0x0005. The State Change Notification Registration Request (SCNReg) message allows an iSNS client to register a Storage Node to receive State Change Notification (SCN) messages.

The SCN notifies the Storage Node of changes to any Storage Nodes within any DD of which it is a member. If the Storage Node is a Control Node, it SHALL receive SCN notifications for changes in the entire network. Note that whereas SCNReg sets the SCN Bitmap field, the DevAttrReg message registers the UDP or TCP Port used by each Portal to receive SCN messages. If no SCN Port fields of any Portals of the Storage Node are registered to receive SCN messages, then the SCNReg message SHALL be rejected with Status Code 17 (SCN Registration Rejected).

The SCNReg request PDU Payload contains a Source Attribute, a Message Key Attribute, and an Operating Attribute. Valid Message Key Attributes for a SCNReg are shown below:

Valid Message Key Attributes for SCNReg
iSCSI Name
FC Port Name WWPN

The node with the iSCSI Name or FC Port Name WWPN attribute that matches the Message Key in the SCNReg message is registered to receive SCNs using the specified SCN bitmap. A maximum of one Node SHALL be registered for each SCNReg message.

The SCN Bitmap is the only operating attribute of this message, and it always overwrites the previous contents of this field in the iSNS database. The bitmap indicates the SCN event types for which the Node is registering.

Note that the settings of this bitmap determine whether the SCN registration is for regular SCNs or management SCNs. Control Nodes MAY conduct registrations for management SCNs; iSNS clients that are not supporting Control Nodes MUST NOT conduct registrations for management SCNs. Control Nodes that register for management SCNs receive a copy of every SCN message generated by the iSNS server. It is recommended that management registrations be used only when needed in order to conserve iSNS server resources. In addition, a Control Node that conducts such registrations should be prepared to receive the anticipated volume of SCN message traffic.

5.6.5.6. SCN Deregister Request (SCNDereg)

The SCNDereg message type is 0x0006. The SCNDereg message allows an iSNS client to stop receiving State Change Notification (SCN) messages.

The SCNDereg request message PDU Payload contains a Source Attribute and Message Key Attribute(s). Valid Message Key Attributes for a SCNDereg are shown below:

Valid Message Key Attributes for SCNDereg
iSCSI Name
FC Port Name WWPN

The node with an iSCSI Name or FC Port Name WWPN attribute that matches the Message Key Attributes in the SCNDereg message is deregistered for SCNs. The SCN bitmap field of such Nodes are cleared. A maximum of one Node SHALL be deregistered for each SCNDereg message.

There are no Operating Attributes in the SCNDereg message.

5.6.5.7. SCN Event (SCNEvent)

The SCNEvent message type is 0x0007. The SCNEvent is a message sent by an iSNS client to request generation of a State Change Notification (SCN) message by the iSNS server. The SCN, sent by the iSNS server, then notifies iFCP, iSCSI, and Control Nodes within the affected DD of the change indicated in the SCNEvent.

Most SCNs are automatically generated by the iSNS server when Nodes are registered or deregistered from the directory database. SCNs are also generated when a network management application or Control Node makes changes to the DD membership in the iSNS server. However, an iSNS client can trigger an SCN by using SCNEvent.

The SCNEvent message PDU Payload contains a Source Attribute, a Message Key Attribute, and an Operating Attribute. Valid Key Attributes for a SCNEvent are shown below:

Valid Message Key Attributes for SCNEvent
iSCSI Name
FC Port Name WWPN

The Operating Attributes section SHALL contain the SCN Event Bitmap attribute. The bitmap indicates the event that caused the SCNEvent to be generated.

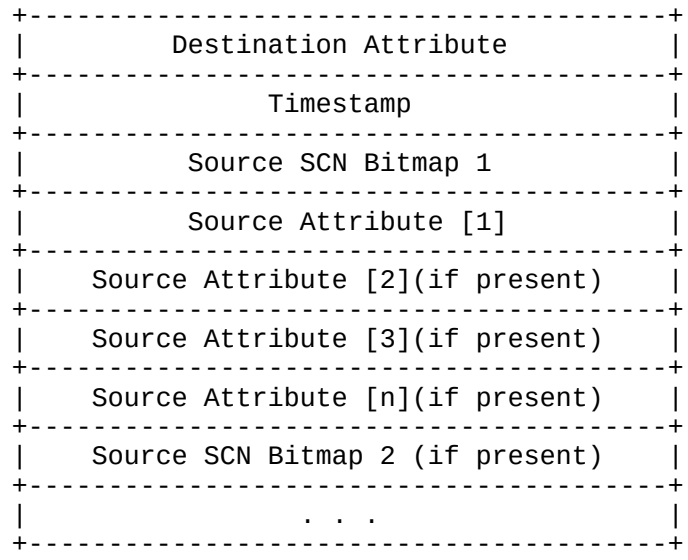
5.6.5.8. State Change Notification (SCN)

The SCN message type is 0x0008. The SCN is a message generated by the iSNS server, notifying a registered Storage Node of changes. There are two types of SCN registrations: regular registrations and management registrations. Regular SCNs notify iSNS clients of events within the discovery domain. Management SCNs notify Control Nodes that register for management SCNs of events occurring anywhere in the network.

If no active TCP connection to the SCN recipient exists, then the SCN message SHALL be sent to one Portal of the registered Storage Node that has a registered TCP or UDP Port value in the SCN Port field. If more than one Portal of the Storage Node has a registered SCN Port value, then the SCN SHALL be delivered to any one of the indicated Portals, provided that the selected Portal is not the subject of the SCN.

The types of events that can trigger an SCN message, and the amount of information contained in the SCN message, depend on the registered SCN Event Bitmap for the Storage Node. The iSCSI Node SCN Bitmap is described in Section 6.4.4. The iFCP SCN Bitmap is described in Section 6.6.12.

The format of the SCN PDU Payload is shown below:



All PDU Payload attributes are in TLV format.

The Destination Attribute is the Node identifier that is receiving the SCN. The Destination Attribute can be an iSCSI Name or FC Port Name.

The Timestamp field, using the Timestamp TLV format, described in Section 6.2.4, indicates the time the SCN was generated.

The Source SCN Bitmap field indicates the type of SCN notification (i.e., regular or management SCN), and the type of event that caused the SCN to be generated; it does not necessarily correlate with the original SCN bitmap registered in the iSNS server.

Following the timestamp, the SCN message SHALL list the SCN bitmap, followed by the key attribute (i.e., iSCSI Name or FC Port Name) of the Storage Node affected by the SCN event. If the SCN is a Management SCN, then the SCN message SHALL also list the DD_ID and/or DDS_ID of the Discovery Domains and Discovery Domain Sets (if any) that caused the change in state for that Storage Node. These additional attributes (i.e., DD_ID and/or DDS_ID) shall immediately follow the iSCSI Name or FC Port Name and precede the next SCN bitmap for the next notification message (if any). The SCN bitmap is used as a delineator for SCN messages providing multiple state change notifications.

For example, a regular SCN for notifying an iSNS client of a new Portal available for a particular iSCSI target would contain the SCN bitmap followed by the iSCSI Name of the target device as the source attribute. If the SCN were a management SCN, then the iSCSI Name would be followed by the DD_ID(s) of the shared Discovery Domains that allow the destination Storage Node to have visibility to the affected Storage Node. If a Discovery Domain Set (DDS) was enabled in order to provide this visibility, then the appropriate DDS_ID would be included as well.

A management SCN is also generated to notify a Control Node of the creation, deletion, or modification of a Discovery Domain or Discovery Domain Set. In this case, the DD_ID and/or DDS_ID of the affected Discovery Domain and/or Discovery Domain Set would follow the SCN bitmap.

For example, a management SCN to notify a Control Node of a new DD within a Discovery Domain Set would contain both the DD_ID and the DDS_ID of the affected Discovery Domain and Discovery Domain Set among the Source Attributes.

See Sections 6.4.4 and 6.6.12 for additional information on the SCN Bitmap.

5.6.5.9. DD Register (DDReg)

The DDReg message type is 0x0009. This message is used to create a new Discovery Domain (DD), to update an existing DD Symbolic Name and/or DD Features attribute, and to add DD members.

DDs are uniquely defined using DD_IDs. DD registration attributes are described in Section 6.11.

The DDReg message PDU Payload contains the Source Attribute and optional Message Key and Operating Attributes.

The Message Key, if used, contains the DD_ID of the Discovery Domain to be registered. If the Message Key contains a DD_ID of an existing DD entry in the iSNS database, then the DDReg message SHALL attempt to update the existing entry. If the DD_ID in the Message Key (if used) does not match an existing DD entry, then the iSNS server SHALL reject the DDReg message with a status code of 3 (Invalid Registration). If the DD_ID is included in both the Message Key and Operating Attributes, then the DD_ID value in the Message Key MUST be the same as the DD_ID value in the Operating Attributes.

A DDReg message with no Message Key SHALL result in the attempted creation of a new Discovery Domain (DD). If the DD_ID attribute (with non-zero length) is included among the Operating Attributes in the DDReg message, then the new Discovery Domain SHALL be assigned the value contained in that DD_ID attribute. Otherwise, if the DD_ID attribute is not contained among the Operating Attributes of the DDReg message, or if the DD_ID is an operating attribute with a TLV length of 0, then the iSNS server SHALL assign a DD_ID value. The assigned DD_ID value is then returned in the DDReg Response message. The Operating Attributes can also contain the DD Member iSCSI Node Index, DD Member iSCSI Name, DD Member FC Port Name, DD Member Portal IP Address, DD Member Portal TCP/UDP Port Number, or DD Member Portal Index of members to be added to the DD. It may also contain the DD_Symbolic_Name and/or DD_Features of the DD.

This message SHALL add any DD members listed as Operating Attributes to the Discovery Domain specified by the DD_ID. If the DD_Features attribute is an Operating Attribute, then it SHALL be stored in the iSNS server as the feature list for the specified DD. If the DD_Symbolic_Name is an operating attribute and its value is unique (i.e., it does not match the registered DD_Symbolic_Name for another DD), then the value SHALL be stored in the iSNS database as the DD_Symbolic_Name for the specified Discovery Domain. If the value for the DD_Symbolic_Name is not unique, then the iSNS server SHALL reject the attempted DD registration with a status code of 3 (Invalid Registration).

When creating a new DD, if the DD_Symbolic_Name is not included in the Operating Attributes, or if it is included with a zero-length TLV, then the iSNS server SHALL provide a unique DD_Symbolic_Name value for the created DD. The assigned DD_Symbolic_Name value SHALL be returned in the DDRegRsp message.

When creating a new DD, if the DD_Features attribute is not included in the Operating Attributes, then the iSNS server SHALL assign the default value. The default value for DD_Features is 0.

DD Member iSCSI Name, DD Member iFCP Node, DD Member Portal IP Address, and DD Member TCP/UDP Port Number attributes included in the Operating Attributes need not match currently existing iSNS database entries. This allows, for example, a Storage Node to be added to a DD even if the Storage Node is not currently registered in the iSNS database. A Storage Node or Portal can thereby be added to a DD at the time of the DDs creation, even if the Storage Node or Portal is not currently active in the storage network.

If the Operating Attributes contain a DD Member iSCSI Name value for a Storage Node that is currently not registered in the iSNS database, then the iSNS server MUST allocate an unused iSCSI Node Index for that Storage Node. The assigned iSCSI Node Index SHALL be returned in the DDRegRsp message as the DD Member iSCSI Node Index. The allocated iSCSI Node Index value SHALL be assigned to the Storage Node if and when it registers in the iSNS database.

If the Operating Attributes contain a DD Member Portal IP Addr and DD Member Portal TCP/UDP value for a Portal that is not currently registered in the iSNS database, then the iSNS server MUST allocate an unused Portal Index value for that Portal. The assigned Portal Index value SHALL be returned in the DDRegRsp message as the DD Member Portal Index. The allocated Portal Index value SHALL be assigned to the Portal if and when it registers in the iSNS database.

DD Member iSCSI Node Index and DD Member Portal Index attributes that are provided in the Operating Attributes MUST match a corresponding iSCSI Node Index or Portal Index of an existing Storage Node or Portal entry in the iSNS database. Furthermore, the DD Member iSCSI Node Index and DD Member Portal Index SHALL NOT be used to add Storage Nodes or Portals to a DD unless those Storage Nodes or Portals are actively registered in the iSNS database.

5.6.5.10. DD Deregister (DDDereg)

The DDDereg message type is 0x000A. This message allows an iSNS client to deregister an existing Discovery Domain (DD) and to remove members from an existing DD.

DDs are uniquely identified using DD_IDs. DD registration attributes are described in Section 6.11.

The DDDereg message PDU Payload contains a Source Attribute, Message Key Attribute, and optional Operating Attributes.

The Message Key Attribute for a DDDereg message is the DD ID for the Discovery Domain being removed or having members removed. If the DD ID matches an existing DD and there are no Operating Attributes, then the DD SHALL be removed and a success Status Code returned. Any existing members of that DD SHALL remain in the iSNS database without membership in the just-removed DD.

If the DD ID matches an existing DD and there are Operating Attributes matching DD members, then the DD members identified by the Operating Attributes SHALL be removed from the DD and a successful Status Code returned.

If a DD Member iSCSI Name identified in the Operating Attributes contains an iSCSI Name for a Storage Node that is not currently registered in the iSNS database or contained in another DD, then the association between that Storage Node and its pre-assigned iSCSI Node Index SHALL be removed. The pre-assigned iSCSI Node Index value no longer has an association to a specific iSCSI Name and can now be re-assigned.

If a DD Member Portal IP Address and DD Member TCP/UDP Port identified in the Operating Attributes reference a Portal that is not currently registered in the iSNS database or contained in another DD, then the association between that Portal and its pre-assigned Portal Index SHALL be removed. The pre-assigned Portal Index value can now be reassigned.

The attempted deregistration of non-existent DD entries SHALL not be considered an error.

5.6.5.11. DDS Register (DDSReg)

The DDSReg message type is 0x000B. This message allows an iSNS client to create a new Discovery Domain Set (DDS), to update an existing DDS Symbolic Name and/or DDS Status, or to add DDS members.

DDSs are uniquely defined using DDS_IDs. DDS registration attributes are described in Section 6.11.1.

The DDSReg message PDU Payload contains the Source Attribute and, optionally, Message Key and Operating Attributes.

The Message Key, if used, contains the DDS_ID of the Discover Domain Set to be registered or modified. If the Message Key contains a DDS_ID of an existing DDS entry in the iSNS database, then the DDSReg message SHALL attempt to update the existing entry. If the DDS_ID in the Message Key (if used) does not match an existing DDS entry, then the iSNS server SHALL reject the DDSReg message with a status code of 3 (Invalid Registration). If the DDS_ID is included in both the Message Key and Operating Attributes, then the DDS_ID value in the Message Key MUST be the same as the DDS_ID value in the Operating Attributes.

A DDSReg message with no Message Key SHALL result in the attempted creation of a new Discovery Domain Set (DDS). If the DDS_ID attribute (with non-zero length) is included among the Operating Attributes in the DDSReg message, then the new Discovery Domain Set SHALL be assigned the value contained in that DDS_ID attribute. Otherwise, if the DDS_ID attribute is not contained among the Operating Attributes of the DDSReg message, or if the DDS_ID is an

operating attribute with a TLV length of 0, then the iSNS server SHALL assign a DDS_ID value. The assigned DDS_ID value is then returned in the DDSReg Response message. The Operating Attributes can also contain the DDS_Symbolic_Name, the DDS Status, and the DD_IDs of Discovery Domains to be added to the DDS.

When creating a new DDS, if the DDS Symbolic Name is included in the Operating Attributes and its value is unique (i.e., it does not match the registered DDS Symbolic Name for another DDS), then the value SHALL be stored in the iSNS database as the DDS Symbolic Name for that DDS. If the value for the DDS Symbolic Name is not unique, then the iSNS server SHALL reject the attempted DDS registration with a status code of 3 (Invalid Registration).

When creating a new DDS, if the DDS Symbolic Name is not included in the Operating Attributes, or if it is included with a zero-length TLV, then the iSNS server SHALL provide a unique DDS Symbolic Name value for the created DDS. The assigned DDS Symbolic Name value SHALL be returned in the DDSRegRsp message.

This message SHALL add any DD_IDs listed as Operating Attributes to the Discovery Domain Set specified by the DDS_ID Message Key Attribute. In addition, if the DDS_Symbolic_Name is an operating attribute and the value is unique, then it SHALL be stored in the iSNS database as the DDS_Symbolic_Name for the specified Discovery Domain Set.

If a DD_ID listed in the Operating Attributes does not match an existing DD, then a new DD using the DD_ID SHALL be created. In this case for the new DD, the iSNS server SHALL assign a unique value for the DD Symbolic Name and SHALL set the DD Features attribute to the default value of 0. These assigned values SHALL be returned in the DDSRegRsp message.

5.6.5.12. DDS Deregister (DDSDereg)

The DDSDereg message type is 0x000C. This message allows an iSNS client to deregister an existing Discovery Domain Set (DDS) or to remove some DDs from an existing DDS.

The DDSDereg message PDU Payload contains a Source Attribute, a Message Key Attribute, and optional Operating Attributes.

The Message Key Attribute for a DDSDereg message is the DDS ID for the DDS being removed or having members removed. If the DDS ID matches an existing DDS and there are no Operating Attributes, then

the DDS SHALL be removed and a success Status Code returned. Any existing members of that DDS SHALL remain in the iSNS database without membership in the just-removed DDS.

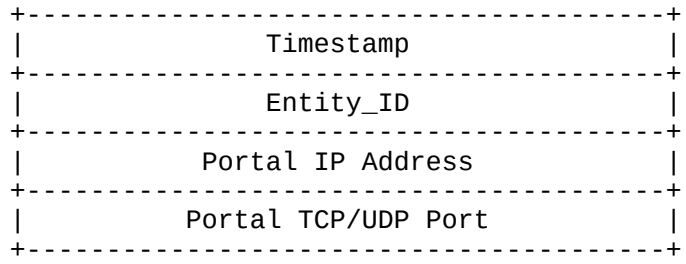
If the DDS ID matches an existing DDS, and there are Operating Attributes matching DDS members, then the DDS members SHALL be removed from the DDS and a success Status Code returned.

The attempted deregistration of non-existent DDS entries SHALL not be considered an error.

5.6.5.13. Entity Status Inquiry (ESI)

The ESI message type is 0x000D. This message is sent by the iSNS server, and is used to verify that an iSNS client Portal is reachable and available. The ESI message is sent to the ESI UDP port provided during registration, or to the TCP connection used for ESI registration, depending on which communication type that is being used.

The ESI message PDU Payload contains the following attributes in TLV format and in the order listed: the current iSNS timestamp, the EID, the Portal IP Address, and the Portal TCP/UDP Port. The format of this message is shown below:



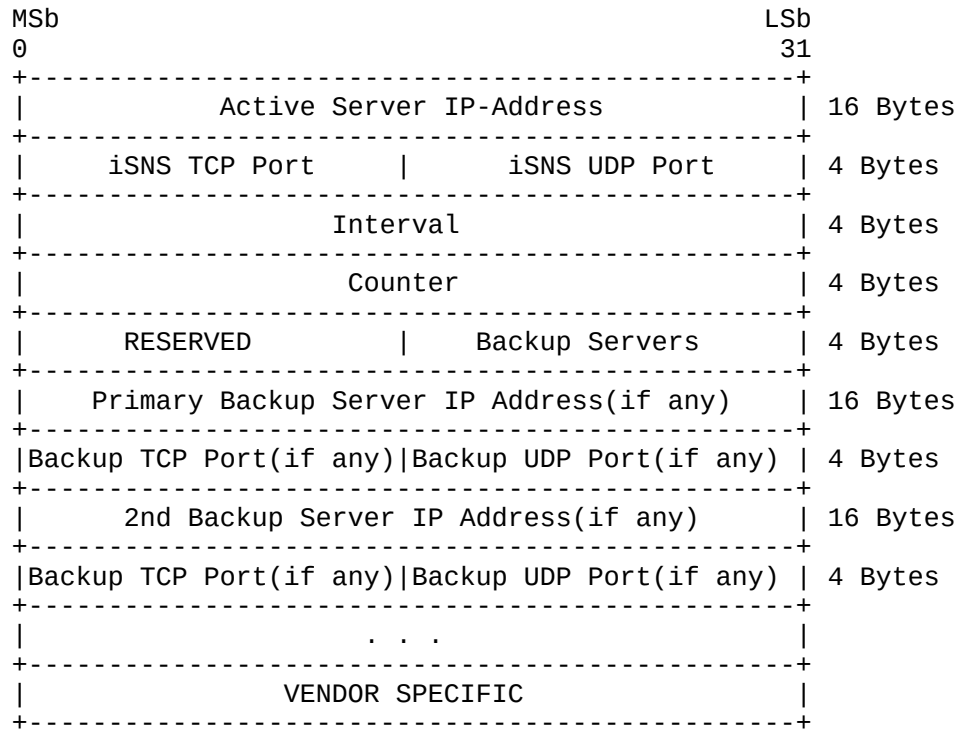
The ESI response message PDU Payload contains a status code, followed by the Attributes from the original ESI message.

If the Portal fails to respond to an administratively-determined number of consecutive ESI messages, then the iSNS server SHALL remove that Portal from the iSNS database. If there are no other remaining ESI-monitored Portals for the associated Network Entity, then the Network Entity SHALL also be removed. The appropriate State Change Notifications, if any, SHALL be triggered.

5.6.5.14. Name Service Heartbeat (Heartbeat)

This message, if used, is only sent by the active iSNS server. It allows iSNS clients and backup servers listening to a broadcast or multicast address to discover the IP address of the primary and backup iSNS servers. It also allows concerned parties to monitor the health and status of the primary iSNS server.

This message is NOT in TLV format. There is no response message to the Name Service Heartbeat.



The heartbeat PDU Payload contains the following:

Active Server IP Address: the IP Address of the active iSNS server in IPv6 format. When this field contains an IPv4 value, it is stored as an IPv4-mapped IPv6 address. That is, the most significant 10 bytes are set to 0x00, with the next two bytes set to 0xFFFF [RFC2373]. When this field contains an IPv6 value, the entire 16-byte field is used.

Active TCP Port: the TCP Port of the server currently in use.

Active UDP Port: the UDP Port of the server currently in use, otherwise 0.

Interval: the interval, in seconds, of the heartbeat.

Counter: a count that begins at 0 when this server becomes active. The count increments by one for each heartbeat sent since this server became active.

Backup Servers: the number of iSNS backup servers. The IP address, TCP Port, and UDP Port of each iSNS backup server follow this field. Note that if backup servers are used, then the active iSNS server SHOULD be among the list of backup servers.

The content of the remainder of this message after the list of backup servers is vendor-specific. Vendors may use additional fields to coordinate between multiple iSNS servers, and/or to identify vendor-specific features.

5.6.5.15. Request FC_DOMAIN_ID (RqstDomId)

The RqstDomId message type is 0x0011. This message is used for iFCP Transparent Mode to allocate non-overlapping FC_DOMAIN_ID values between 1 and 239. The iSNS server becomes the address assignment authority for the entire iFCP fabric. To obtain multiple FC_DOMAIN_ID values, this request must be repeated to the iSNS server multiple times. iSNS clients that acquire FC_DOMAIN_ID values from an iSNS server MUST register for ESI monitoring from that iSNS server.

The RqstDomId PDU Payload contains three TLV attributes in the following order: the requesting Switch Name (WWN) as the Source Attribute, the Virtual_Fabric_ID as the Message Key Attribute, and Preferred ID as the operating attribute. The Virtual_Fabric_ID is a string identifying the domain space for which the iSNS server SHALL allocate non-overlapping integer FC_DOMAIN_ID values between 1 and 239. The Preferred_ID is the nominal FC_DOMAIN_ID value requested by the iSNS client. If the Preferred_ID value is available and has not already been allocated for the Virtual_Fabric_ID specified in the message, the iSNS server SHALL return the requested Preferred_ID value as the Assigned_ID to the requesting client.

The RqstDomId response contains a Status Code, and the TLV attribute Assigned ID, which contains the integer value in the space requested. If no further unallocated values are available from this space, the iSNS server SHALL respond with the Status Code 18 "FC_DOMAIN_ID Not Available".

Once a FC_DOMAIN_ID value has been allocated to an iSNS client by the iSNS server for a given Virtual_Fabric_ID, that FC_DOMAIN_ID value SHALL NOT be reused until it has been deallocated, or until ESI monitoring detects that the iSNS client no longer exists on the network and objects for that client are removed from the iSNS database.

The iSNS server and client SHALL use TCP to transmit and receive RqstDomId, RqstDomIdRsp, RlseDomId, and RlseDomIdRsp messages.

5.6.5.16. Release FC_DOMAIN_ID (RlseDomId)

The RlseDomId message type is 0x0012. This message may be used by iFCP Transparent Mode to release integer identifier values used to assign 3-byte Fibre Channel PORT_ID values.

The RlseDomId message contains three TLV attributes in the following order: the requesting EID as the Source Attribute, the Virtual_Fabric_ID as the Message Key Attribute, and Assigned_ID as the operating attribute. Upon receiving the RlseDomId message, the iSNS server SHALL deallocate the FC_DOMAIN_ID value contained in the Assigned_ID attribute for the Virtual_Fabric_ID attribute specified. Upon deallocation, that FC_DOMAIN_ID value can then be requested by and assigned to a different iSNS client.

The iSNS server and client SHALL use TCP to transmit and receive RqstDomId, RqstDomIdRsp, RlseDomId, and RlseDomIdRsp messages.

5.6.5.17. Get FC_DOMAIN_IDs (GetDomId)

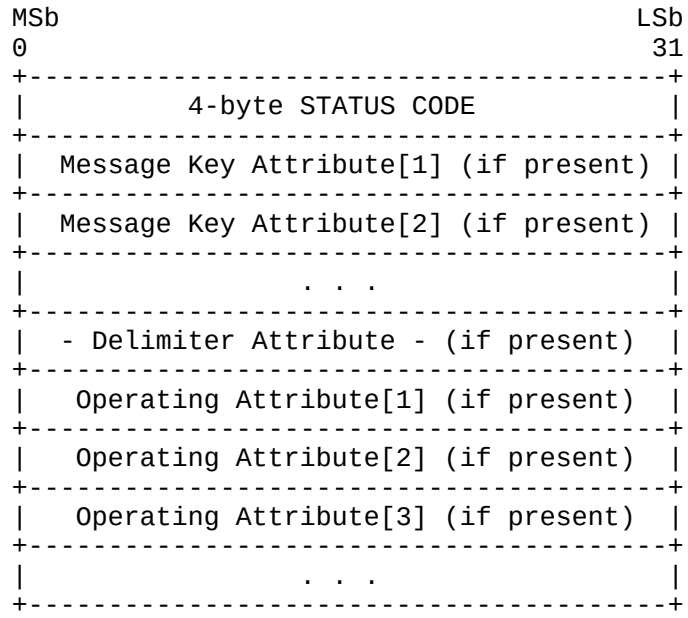
The GetDomId message type is 0x0013. This message is used to learn the currently-allocated FC_DOMAIN_ID values for a given Virtual_Fabric_ID.

The GetDomId message PDU Payload contains a Source Attribute and Message Key Attribute.

The Message Key Attribute for the GetDomId message is the Virtual_Fabric_ID. The response to this message returns all the FC_DOMAIN_ID values that have been allocated for the Virtual_Fabric_ID specified.

5.7. Messages

The iSNSP response message PDU Payloads contain a Status Code, followed by a list of attributes, and have the following format:



The iSNSP Response messages SHALL be sent to the iSNS Client IP Address and the originating TCP/UDP Port that was used for the associated registration and query message.

5.7.1. Status Code

The first field in an iSNSP response message PDU Payload is the Status Code for the operation that was performed. The Status Code encoding is defined in Section 5.4.

5.7.2. Message Key Attributes in Response

Depending on the specific iSNSP request, the response message MAY contain Message Key Attributes. Message Key Attributes generally contain the interesting key attributes that are affected by the operation specified in the original iSNS registration or query message.

5.7.3. Delimiter Attribute in Response

The Delimiter Attribute separates the key and Operating Attributes in a response message, if they exist. The Delimiter Attribute has a tag value of 0 and a length value of 0. The Delimiter Attribute is effectively 8 bytes long: a 4-byte tag containing 0x00000000, and a 4 Byte length field containing 0x00000000.

5.7.4. Operating Attributes in Response

The Operating Attributes in a response are the results related to the iSNS registration or query operation being performed. Some response messages will not have Operating Attributes.

5.7.5. Registration and Query Response Message Types

The following sections describe each query and message type.

5.7.5.1. Device Attribute Registration Response (DevAttrRegRsp)

The DevAttrRegRsp message type is 0x8001. The DevAttrRegRsp message contains the results for the DevAttrReg message with the same TRANSACTION ID.

The Message Key in the DevAttrRegRsp message SHALL return the Message Key in the original registration message. If the iSNS server assigned the Entity Identifier for a Network Entity, then the Message Key Attribute field SHALL contain the assigned Entity Identifier.

The Operating Attributes of the DevAttrRegRsp message SHALL contain the affected object's key and non-key attributes that have been explicitly modified or created by the original DevAttrReg message. Among the Operating Attributes, each modified or added non-key attribute SHALL be listed after its key attribute(s) in the DevAttrRegRsp message. Implicitly registered attributes MUST NOT be returned in the DevAttrRegRsp message. Implicitly registered attributes are those that are assigned a fixed default value or secondary index value by the iSNS server.

Implicitly registered PG objects (i.e., PG objects that are not explicitly included in the registration or replace message) MUST NOT have their key or non-key attributes returned in the DevAttrRegRsp message. However, explicitly registered PG objects (i.e., those with PGT values that are explicitly included in the registration or replace message) SHALL have their PGT values returned in the DevAttrRegRsp message.

For example, three Portals are registered in the original DevAttrReg request message. Due to lack of resources, the iSNS server needs to modify the registered ESI Interval value of one of those Portals. To accomplish this, the iSNS server returns the key attributes identifying the Portal, followed by the non-key modified ESI Interval attribute value, as Operating Attributes of the corresponding DevAttrRegRsp message.

If the iSNS server rejects a registration due to invalid attribute values or types, then the indicated status code SHALL be 3 (Invalid Registration). If this occurs, then the iSNS server MAY include the list of invalid attributes in the Operating Attributes of the DevAttrRsp message.

Some attributes values (e.g., ESI Interval, Registration Period) in the original registration message MAY be modified by the iSNS server. This can occur only for a limited set of attribute types, as indicated in the table in Section 6.1. When this occurs, the registration SHALL be considered a success (with status code 0), and the changed value(s) indicated in the Operating Attributes of the DevAttrRsp message.

5.7.5.2. Device Attribute Query Response (DevAttrQryRsp)

The DevAttrQryRsp message type is 0x8002. The DevAttrQryRsp message contains the results for the DevAttrQry message with the same TRANSACTION ID.

The Message Key in the DevAttrQryRsp message SHALL return the Message Key in the original query message.

If no Operating Attributes are included in the original query, then all Operating Attributes SHALL be returned in the response.

For a successful query result, the DevAttrQryRsp Operating Attributes SHALL contain the results of the original DevAttrQry message.

5.7.5.3. Device Get Next Response (DevGetNextRsp)

The DevGetNextRsp message type is 0x8003. The DevGetNextRsp message contains the results for the DevGetNext message with the same TRANSACTION ID.

The Message Key Attribute field returns the object keys for the next object after the Message Key Attribute in the original DevGetNext message.

The Operating Attribute field returns the Operating Attributes of the next object as requested in the original DevGetNext message. The values of the Operating Attributes are those associated with the object identified by the Message Key Attribute field of the DevGetNextRsp message.

5.7.5.4. Deregister Device Response (DevDeregRsp)

The DevDeregRsp message type is 0x8004. This message is the response to the DevDereg request message.

This message response does not contain a Message Key, but MAY contain Operating Attributes.

In the event of an error, this response message contains the appropriate status code as well as a list of objects from the original DevDereg message that were not successfully deregistered from the iSNS database. This list of objects is contained in the Operating Attributes of the DevDeregRsp message. Note that an attempted deregistration of a non-existent object does not constitute an error, and non-existent entries SHALL not be returned in the DevDeregRsp message.

5.7.5.5. SCN Register Response (SCNRegRsp)

The SCNRegRsp message type is 0x8005. This message is the response to the SCNReg request message.

The SCNRegRsp message does not contain any Message Key or Operating Attributes.

5.7.5.6. SCN Deregister Response (SCNDeregRsp)

The SCNDeregRsp message type is 0x8006. This message is the response to the SCNDereg request message.

The SCNDeregRsp message does not contain any Message Key or Operating Attributes.

5.7.5.7. SCN Event Response (SCNEventRsp)

The SCNEventRsp message type is 0x8007. This message is the response to the SCNEvent request message.

The SCNEventRsp message does not contain any Message Key or Operating Attributes.

5.7.5.8. SCN Response (SCNRsp)

The SCNRsp message type is 0x8008. This message is sent by an iSNS client, and provides confirmation that the SCN message was received and processed.

The SCNRsp response contains the SCN Destination Attribute representing the Node identifier that received the SCN.

5.7.5.9. DD Register Response (DDRegRsp)

The DDRegRsp message type is 0x8009. This message is the response to the DDReg request message.

The Message Key in the DDRegRsp message SHALL return the Message Key in the original query message. If the original DDReg message did not have a Message Key, then the DDRegRsp message SHALL not have a Message Key.

If the DDReg operation is successful, the DD ID of the DD created or updated SHALL be returned as an operating attribute of the message.

If the DD Symbolic Name attribute or DD Features attribute was assigned or updated during the DDReg operation, then any new values SHALL be returned as an operating attribute of the DDRegRsp message.

If the iSNS server rejects a DDReg due to invalid attribute values or types, then the indicated status code SHALL be 3 (Invalid Registration). If this occurs, then the iSNS server MAY include the list of invalid attributes in the Operating Attributes of the DDRegRsp message.

5.7.5.10. DD Deregister Response (DDDeregRsp)

The DDDeregRsp message type is 0x800A. This message is the response to the DDDereg request message.

The DDDeregRsp message does not contain any Message Key or Operating Attributes.

5.7.5.11. DDS Register Response (DDSRegRsp)

The DDSRegRsp message type is 0x800B. This message is the response to the DDSReg request message.

The Message Key in the DDSRegRsp message SHALL contain the Message Key of the original DDSReg message. If the original DDSReg message did not have a Message Key, then the DDSRegRsp message SHALL NOT have a Message Key.

If the DDSReg operation is successful, the DDS ID of the DDS created or updated SHALL be returned as an operating attribute of the message.

If the DDS Symbolic Name attribute or DDS Status attribute was assigned or updated during the DDSRegRsp operation, then any new values SHALL be returned as an operating attribute of the DDSRegRsp message.

If the iSNS server rejects a DDSReg due to invalid attribute values or types, then the indicated status code SHALL be 3 (Invalid Registration). If this occurs, then the iSNS server MAY include the list of invalid attributes in the Operating Attributes of the DDSRegRsp message.

5.7.5.12. DDS Deregister Response (DDSDeregRsp)

The DDSDeregRsp message type is 0x800C. This message is the response to the DDSDereg request message.

The DDSDeregRsp message does not contain any Message Key or Operating Attributes.

5.7.5.13. Entity Status Inquiry Response (ESIRsp)

The ESIRsp message type is 0x800D. This message is sent by an iSNS client and provides confirmation that the ESI message was received and processed.

The ESIRsp response message PDU Payload contains the attributes from the original ESI message. These attributes represent the Portal that is responding to the ESI. The ESIRsp Attributes are in the order they were provided in the original ESI message.

Upon receiving the ESIRsp from the iSNS client, the iSNS server SHALL update the timestamp attribute for that Network Entity and Portal.

5.7.5.14. Request FC_DOMAIN_ID Response (RqstDomIdRsp)

The RqstDomIdRsp message type is 0x8011. This message provides the response for RqstDomId.

The RqstDomId response contains a Status Code and the TLV attribute Assigned ID, which contains the integer value in the space requested. If no further unallocated values are available from this space, the iSNS server SHALL respond with the Status Code 19 "FC_DOMAIN_ID Not Available".

Once a FC_DOMAIN_ID value is allocated by the iSNS server, it SHALL NOT be reused until it has been deallocated by the iSNS client to which the value was assigned, or until the ESI message detects that the iSNS client no longer exists on the network.

The iSNS server and client SHALL use TCP to transmit and receive RqstDomId, RqstDomIdRsp, RlseDomId, and RlseDomIdRsp messages.

5.7.5.15. Release FC_DOMAIN_ID Response (RlseDomIdRsp)

The RlseDomIdRsp message type is 0x8012. This message provides the response for RlseDomId. The response contains an Error indicating whether the request was successful. If the Assigned_ID value in the original RlseDomId message is not allocated, then the iSNS server SHALL respond with this message using the Status Code 20 "FC_DOMAIN_ID Not Allocated".

The iSNS server and client SHALL use TCP to transmit and receive RqstDomId, RqstDomIdRsp, RlseDomId, and RlseDomIdRsp messages.

5.7.5.16. Get FC_DOMAIN_IDs Response (GetDomIdRsp)

The GetDomIdRsp message type is 0x8013. This message is used to determine which FC_DOMAIN_ID values have been allocated for the Virtual_Fabric_ID specified in the original GetDomId request message.

The GetDomId response message PDU Payload contains a Status Code indicating whether the request was successful, and a list of the Assigned IDs from the space requested. The Assigned_ID attributes are listed in TLV format.

5.8. Vendor-Specific Messages

Vendor-specific iSNSP messages have a functional ID of between 0x0100 and 0x01FF, whereas vendor-specific responses have a functional ID of between 0x8100 and 0x81FF. The first Message Key Attribute in a

vendor-specific message SHALL be the company OUI (tag=256) identifying the original creator of the proprietary iSNSP message. The contents of the remainder of the message are vendor-specific.

6. iSNS Attributes

Attributes can be stored in the iSNS server using iSNSP registration messages, and they can be retrieved using iSNSP query messages. Unless otherwise indicated, these attributes are supplied by iSNS clients using iSNSP registration messages.

6.1. iSNS Attribute Summary

The complete registry of iSNS attributes is maintained by IANA, and the following table summarizes the initial set of iSNS attributes available at the time of publication of this document.

Attributes	Length	Tag	Reg Key	Query Key
Delimiter	0	0	N/A	N/A
Entity Identifier (EID)	4-256	1	1	1|2|16&17|32|64
Entity Protocol	4	2	1	1|2|16&17|32|64
Management IP Address	16	3	1	1|2|16&17|32|64
Timestamp	8	4	--	1|2|16&17|32|64
Protocol Version Range	4	5	1	1|2|16&17|32|64
Registration Period	4	6	1	1|2|16&17|32|64
Entity Index	4	7	1	1|2|16&17|32|64
Entity Next Index	4	8	--	1|2|16&17|32|64
Entity ISAKMP Phase-1	var	11	1	1|2|16&17|32|64
Entity Certificate	var	12	1	1|2|16&17|32|64
Portal IP Address	16	16	1	1|16&17|32|64
Portal TCP/UDP Port	4	17	1	1|16&17|32|64
Portal Symbolic Name	4-256	18	16&17	1|16&17|32|64
ESI Interval	4	19	16&17	1|16&17|32|64
ESI Port	4	20	16&17	1|16&17|32|64
Portal Index	4	22	16&17	1|16&17|32|64
SCN Port	4	23	16&17	1|16&17|32|64
Portal Next Index	4	24	--	1|16&17|32|64
Portal Security Bitmap	4	27	16&17	1|16&17|32|64
Portal ISAKMP Phase-1	var	28	16&17	1|16&17|32|64
Portal ISAKMP Phase-2	var	29	16&17	1|16&17|32|64
Portal Certificate	var	31	16&17	1|16&17|32|64
iSCSI Name	4-224	32	1	1|16&17|32|33
iSCSI Node Type	4	33	32	1|16&17|32
iSCSI Alias	4-256	34	32	1|16&17|32
iSCSI SCN Bitmap	4	35	32	1|16&17|32
iSCSI Node Index	4	36	32	1|16&17|32
WWNN Token	8	37	32	1|16&17|32

iSCSI Node Next Index	4	38	--	1|16&17|32
iSCSI AuthMethod	var	42	32	1|16&17|32
PG iSCSI Name	4-224	48	32|16&17	1|16&17|32|52
PG Portal IP Addr	16	49	32|16&17	1|16&17|32|52
PG Portal TCP/UDP Port	4	50	32|16&17	1|16&17|32|52
PG Tag (PGT)	4	51	32|16&17	1|16&17|32|52
PG Index	4	52	32|16&17	1|16&17|32|52
PG Next Index	4	53	--	1|16&17|32|52
FC Port Name WWPN	8	64	1	1|16&17|64|66|96|128
Port ID	4	65	64	1|16&17|64
FC Port Type	4	66	64	1|16&17|64
Symbolic Port Name	4-256	67	64	1|16&17|64
Fabric Port Name	8	68	64	1|16&17|64
Hard Address	4	69	64	1|16&17|64
Port IP-Address	16	70	64	1|16&17|64
Class of Service	4	71	64	1|16&17|64
FC-4 Types	32	72	64	1|16&17|64
FC-4 Descriptor	4-256	73	64	1|16&17|64
FC-4 Features	128	74	64	1|16&17|64
iFCP SCN bitmap	4	75	64	1|16&17|64
Port Role	4	76	64	1|16&17|64
Permanent Port Name	8	77	--	1|16&17|64
FC-4 Type Code	4	95	--	1|16&17|64
FC Node Name WWNN	8	96	64	1|16&17|64|96
Symbolic Node Name	4-256	97	96	64|96
Node IP-Address	16	98	96	64|96
Node IPA	8	99	96	64|96
Proxy iSCSI Name	4-256	101	96	64|96
Switch Name	8	128	128	128
Preferred ID	4	129	128	128
Assigned ID	4	130	128	128
Virtual_Fabric_ID	4-256	131	128	128
iSNS Server Vendor OUI	4	256	--	SOURCE Attribute
Vendor-Spec iSNS Srvr		257-384	--	SOURCE Attribute
Vendor-Spec Entity		385-512	1	1|2|16&17|32|64
Vendor-Spec Portal		513-640	16&17	1|16&17|32|64
Vendor-Spec iSCSI Node		641-768	32	16&17|32
Vendor-Spec FC Port Name		769-896	64	1|16&17|64
Vendor-Spec FC Node Name		897-1024	96	64|96
Vendor-Specific DDS		1025-1280	2049	2049
Vendor-Specific DD		1281-1536	2065	2065
Other Vendor-Specific		1537-2048		
DD_Set ID	4	2049	2049	1|32|64|2049|2065
DD_Set Sym Name	4-256	2050	2049	2049
DD_Set Status	4	2051	2049	2049
DD_Set_Next_ID	4	2052	--	2049
DD_ID	4	2065	2049	1|32|64|2049|2065
DD_Symbolic Name	4-256	2066	2065	2065

DD_Member iSCSI Index	4	2067	2065	2065
DD_Member iSCSI Name	4-224	2068	2065	2065
DD_Member FC Port Name	8	2069	2065	2065
DD_Member Portal Index	4	2070	2065	2065
DD_Member Portal IP Addr	16	2071	2065	2065
DD_Member Portal TCP/UDP	4	2072	2065	2065
DD_Features	4	2078	2065	2065
DD_ID Next ID	4	2079	--	2065

The following are descriptions of the columns used in the above table:

Length: indicates the attribute length in bytes used for the TLV format. Variable-length identifiers are NULL-terminated and 4-byte aligned (NULLs are included in the length).

Tag: the IANA-assigned integer tag value used to identify the attribute. All undefined tag values are reserved.

Reg Key: indicates the tag values for the object key in DevAttrReg messages for registering a new attribute value in the database. These tags represent attributes defined as object keys in Section 4.

Query Key: indicates the possible tag values for the Message Key and object key that are used in the DevAttrQry messages for retrieving a stored value from the iSNS database.

The following is a summary of iSNS attribute tag values available for future allocation by IANA at the time of publication:

Tag Values	Reg Key	Query Key
9-10, 13-15	1	1|2|16&17|32|64
21, 25-26, 30	16&17	1|16&17|32|64
39-41, 44-47	32	1|16&17|32
54-63	32|16&17	1|16&17|32|52
78-82, 85-94	64	1|16&17|64
102-127	96	64|96
132-255	--	SOURCE Attribute
2053-2064	2049	2049
2073-2077	2065	2065
2080-65535	To be assigned	To be assigned

Registration and query keys for attributes with tags in the range 2080 to 65535 are to be documented in the RFC introducing the new iSNS attributes. IANA will maintain registration of these values as required by the new RFC.

New iSNS attributes with any of the above tag values MAY also be designated as "read-only" attributes. The new RFC introducing these attributes as "read-only" SHALL document them as such, and IANA will record their corresponding Registration Keys (Reg Keys) as "--".

6.2. Entity Identifier-Keyed Attributes

The following attributes are stored in the iSNS server using the Entity Identifier attribute as the key.

6.2.1. Entity Identifier (EID)

The Entity Identifier (EID) is variable-length UTF-8 encoded NULL-terminated text-based description for a Network Entity. This key attribute uniquely identifies each Network Entity registered in the iSNS server. The attribute length varies from 4 to 256 bytes (including the NULL termination), and is a unique value within the iSNS server.

If the iSNS client does not provide an EID during registration, the iSNS server SHALL generate one that is unique within the iSNS database. If an EID is to be generated, then the EID attribute value in the registration message SHALL be empty (0 length). The generated EID SHALL be returned in the registration response.

In environments where the iSNS server is integrated with a DNS infrastructure, the Entity Identifier may be used to store the Fully Qualified Domain Name (FQDN) of the iSCSI or iFCP device. FQDNs of greater than 255 bytes MUST NOT be used.

If FQDNs are not used, the iSNS server can be used to generate EIDs. EIDs generated by the iSNS server MUST begin with the string "isns:". iSNS clients MUST NOT generate and register EIDs beginning with the string "isns:".

This field MUST be normalized according to the nameprep template [NAMEPREP] before it is stored in the iSNS database.

6.2.2. Entity Protocol

The Entity Protocol is a required 4-byte integer attribute that indicates the block storage protocol used by the registered NETWORK ENTITY. Values used for this attribute are assigned and maintained by IANA. The initial set of protocols supported by iSNS is as follows:

Value	Entity Protocol Type
1	No Protocol
2	iSCSI
3	iFCP
All others	To be assigned by IANA

'No Protocol' is used to indicate that the Network Entity does not support an IP block storage protocol. A Control Node or monitoring Node would likely (but not necessarily) use this value.

This attribute is required during initial registration of the Network Entity.

6.2.3. Management IP Address

This field contains the IP Address that may be used to manage the Network Entity and all Storage Nodes contained therein via the iSNS MIB [iSNSMIB]. Some implementations may also use this IP address to support vendor-specific proprietary management protocols. The Management IP Address is a 16-byte field that may contain an IPv4 or IPv6 address. When this field contains an IPv4 value, it is stored as an IPv4-mapped IPv6 address. That is, the most significant 10 bytes are set to 0x00, with the next two bytes set to 0xFFFF [RFC2373]. When this field contains an IPv6 value, the entire 16-byte field is used. If this field is not set, then in-band management through the IP address of one of the Portals of the Network Entity is assumed.

6.2.4. Entity Registration Timestamp

This field indicates the most recent time when the Network Entity registration occurred or when an associated object attribute was updated or queried by the iSNS client registering the Network Entity. The time format is, in seconds, the update period since the standard base time of 00:00:00 GMT on January 1, 1970. This field cannot be explicitly registered. This timestamp TLV format is also used in the SCN and ESI messages.

6.2.5. Protocol Version Range

This field contains the minimum and maximum version of the block storage protocol supported by the Network Entity. The most significant two bytes contain the maximum version supported, and the least significant two bytes contain the minimum version supported. If a range is not registered, then the Network Entity is assumed to

support all versions of the protocol. The value 0xffff is a wildcard that indicates no minimum or maximum. If the Network Entity does not support a protocol, then this field SHALL be set to 0.

6.2.6. Registration Period

This 4-byte unsigned integer field indicates the maximum period, in seconds, that the registration SHALL be maintained by the server without receipt of an iSNS message from the iSNS client that registered the Network Entity. Entities that are not registered for ESI monitoring MUST have a non-zero Registration Period. If a Registration Period is not requested by the iSNS client and Entity Status Inquiry (ESI) messages are not enabled for that client, then the Registration Period SHALL be set to a non-zero value by the iSNS server. This implementation-specific value for the Registration Period SHALL be returned in the registration response to the iSNS client. The Registration Period may be set to zero, indicating its non-use, only if ESI messages are enabled for that Network Entity.

The registration SHALL be removed from the iSNS database if an iSNS Protocol message is not received from the iSNS client before the registration period has expired. Receipt of any iSNS Protocol message from the iSNS client automatically refreshes the Entity Registration Period and Entity Registration Timestamp. To prevent a registration from expiring, the iSNS client should send an iSNS Protocol message to the iSNS server at intervals shorter than the registration period. Such a message can be as simple as a query for one of its own attributes, using its associated iSCSI Name or FC Port Name WWPN as the Source attribute.

For an iSNS client that is supporting a Network Entity with multiple Storage Node objects, receipt of an iSNS message from any Storage Node of that Network Entity is sufficient to refresh the registration for all Storage Node objects of the Network Entity.

If ESI support is requested as part of a Portal registration, the ESI Response message received from the iSNS client by the iSNS server SHALL refresh the registration.

6.2.7. Entity Index

The Entity Index is an unsigned non-zero integer value that uniquely identifies each Network Entity registered in the iSNS server. Upon initial registration of a Network Entity, the iSNS server assigns an unused value for the Entity Index. Each Network Entity in the iSNS database MUST be assigned a value for the Entity Index that is not

assigned to any other Network Entity. Furthermore, Entity Index values for recently deregistered Network Entities SHOULD NOT be reused in the short term.

The Entity Index MAY be used to represent the Network Entity in situations when the Entity Identifier is too long or otherwise inappropriate. An example of this is when SNMP is used for management, as described in Section 2.10.

6.2.8. Entity Next Index

This is a virtual attribute containing a 4-byte integer value that indicates the next available (i.e., unused) Entity Index value. This attribute may only be queried; the iSNS server SHALL return an error code of 3 (Invalid Registration) to any client that attempts to register a value for this attribute. A Message Key is not required when exclusively querying for this attribute.

The Entity Next Index MAY be used by an SNMP client to create an entry in the iSNS server. SNMP requirements are described in Section 2.10.

6.2.9. Entity ISAKMP Phase-1 Proposals

This field contains the IKE Phase-1 proposal, listing in decreasing order of preference the protection suites acceptable to protect all IKE Phase-2 messages sent and received by the Network Entity. This includes Phase-2 SAs from the iSNS client to the iSNS server as well as to peer iFCP and/or iSCSI devices. This attribute contains the SA payload, proposal payload(s), and transform payload(s) in the ISAKMP format defined in [RFC2408].

This field should be used if the implementer wishes to define a single phase-1 SA security configuration used to protect all phase-2 IKE traffic. If the implementer desires to have a different phase-1 SA security configuration to protect each Portal interface, then the Portal Phase-1 Proposal (Section 6.3.10) should be used.

6.2.10. Entity Certificate

This attribute contains one or more X.509 certificates that are bound to the Network Entity. This certificate is uploaded and registered to the iSNS server by clients wishing to allow other clients to authenticate themselves and to access the services offered by that Network Entity. The format of the X.509 certificate is found in [RFC3280]. This certificate MUST contain a Subject Name with an empty sequence and MUST contain a SubjectAltName extension encoded

with the dNSName type. The Entity Identifier (Section 6.2.1) of the identified Entity MUST be stored in the SubjectAltName field of the certificate.

6.3. Portal-Keyed Attributes

The following Portal attributes are registered in the iSNS database using the combined Portal IP-Address and Portal TCP/UDP Port as the key. Each Portal is associated with one Entity Identifier object key.

6.3.1. Portal IP Address

This attribute is the IP address of the Portal through which a Storage Node can transmit and receive storage data. The Portal IP Address is a 16-byte field that may contain an IPv4 or IPv6 address. When this field contains an IPv4 address, it is stored as an IPv4-mapped IPv6 address. That is, the most significant 10 bytes are set to 0x00, with the next 2 bytes set to 0xFFFF [RFC2373]. When this field contains an IPv6 address, the entire 16-byte field is used. The Portal IP Address and the Portal TCP/UDP Port number (see 6.3.2 below) are used as a key to identify a Portal uniquely. It is a required attribute for registration of a Portal.

6.3.2. Portal TCP/UDP Port

The TCP/UDP port of the Portal through which a Storage Node can transmit and receive storage data. Bits 16 to 31 represents the TCP/UDP port number. Bit 15 represents the port type. If bit 15 is set, then the port type is UDP. Otherwise it is TCP. Bits 0 to 14 are reserved.

If the field value is 0, then the port number is the implied canonical port number and type of the protocol indicated by the associated Entity Type.

The Portal IP Address and the Portal TCP/UDP Port number are used as a key to identify a Portal uniquely. It is a required attribute for registration of a Portal.

6.3.3. Portal Symbolic Name

A variable-length UTF-8 encoded NULL-terminated text-based description of up to 256 bytes. The Portal Symbolic Name is a user-readable description of the Portal entry in the iSNS server.

6.3.4. Entity Status Inquiry Interval

This field indicates the requested time, in seconds, between Entity Status Inquiry (ESI) messages sent from the iSNS server to this Network Entity. ESI messages can be used to verify that a Portal registration continues to be valid. To request monitoring by the iSNS server, an iSNS client registers a non-zero value for this Portal attribute using a DevAttrReg message. The client MUST register an ESI Port on at least one of its Portals to receive the ESI monitoring.

If the iSNS server does not receive an expected response to an ESI message, it SHALL attempt an administratively configured number of re-transmissions of the ESI message. The ESI Interval period begins with the iSNS server's receipt of the last ESI Response. All re-transmissions MUST be sent before twice the ESI Interval period has passed. If no response is received from any of the ESI messages, then the Portal SHALL be deregistered. Note that only Portals that have registered a value in their ESI Port field can be deregistered in this way.

If all Portals associated with a Network Entity that have registered for ESI messages are deregistered due to non-response, and if no registrations have been received from the client for at least two ESI Interval periods, then the Network Entity and all associated objects (including Storage Nodes) SHALL be deregistered.

If the iSNS server is unable to support ESI messages or the ESI Interval requested, it SHALL either reject the ESI request by returning an "ESI Not Available" Status Code or modify the ESI Interval attribute by selecting its own suitable value and returning that value in the Operating Attributes of the registration response message.

If at any time an iSNS client that is registered for ESI messages has not received an ESI message to any of its Portals as expected, then the client MAY attempt to query the iSNS server using a DevAttrQry message using its Entity_ID as the key. If the query result is the error "no such entry", then the client SHALL close all remaining TCP connections to the iSNS server and assume that it is no longer registered in the iSNS database. Such a client MAY attempt re-registration.

6.3.5. ESI Port

This field contains the TCP or UDP port used for ESI monitoring by the iSNS server at the Portal IP Address. Bits 16 to 31 represent the port number. If bit 15 is set, then the port type is UDP. Otherwise, the port is TCP. Bits 0 to 14 are reserved.

If the iSNS client registers a valid TCP or UDP port number in this field, then the client SHALL allow ESI messages to be received at the indicated TCP or UDP port. If a TCP port is registered and a pre-existing TCP connection from that TCP port to the iSNS server does not already exist, then the iSNS client SHALL accept new TCP connections from the iSNS server at the indicated TCP port.

The iSNS server SHALL return an error if a Network Entity is registered for ESI monitoring and none of the Portals of that Network Entity has an entry for the ESI Port field. If multiple Portals have a registered ESI port, then the ESI message may be delivered to any one of the indicated Portals.

6.3.6. Portal Index

The Portal Index is a 4-byte non-zero integer value that uniquely identifies each Portal registered in the iSNS database. Upon initial registration of a Portal, the iSNS server assigns an unused value for the Portal Index of that Portal. Each Portal in the iSNS database MUST be assigned a value for the Portal Index that is not assigned to any other Portal. Furthermore, Portal Index values for recently deregistered Portals SHOULD NOT be reused in the short term.

The Portal Index MAY be used to represent a registered Portal in situations where the Portal IP-Address and Portal TCP/UDP Port is unwieldy to use. An example of this is when SNMP is used for management, as described in Section 2.10.

6.3.7. SCN Port

This field contains the TCP or UDP port used by the iSNS client to receive SCN messages from the iSNS server. When a value is registered for this attribute, an SCN message may be received on the indicated port for any of the Storage Nodes supported by the Portal. Bits 16 to 31 contain the port number. If bit 15 is set, then the port type is UDP. Otherwise, the port type is TCP. Bits 0 to 14 are reserved.

If the iSNS client registers a valid TCP or UDP port number in this field, then the client SHALL allow SCN messages to be received at the indicated TCP or UDP port. If a TCP port is registered and a pre-

existing TCP connection from that TCP port to the iSNS server does not already exist, then the iSNS client SHALL accept new TCP connections from the iSNS server at the indicated TCP port.

The iSNS server SHALL return an error if an SCN registration message is received and none of the Portals of the Network Entity has an entry for the SCN Port. If multiple Portals have a registered SCN Port, then the SCN SHALL be delivered to any one of the indicated Portals of that Network Entity.

6.3.8. Portal Next Index

This is a virtual attribute containing a 4-byte integer value that indicates the next available (i.e., unused) Portal Index value. This attribute may only be queried; the iSNS server SHALL return an error code of 3 (Invalid Registration) to any client that attempts to register a value for this attribute. A Message Key is not required when exclusively querying for this attribute.

The Portal Next Index MAY be used by an SNMP client to create an entry in the iSNS server. SNMP requirements are described in Section 2.10.

6.3.9. Portal Security Bitmap

This 4-byte field contains flags that indicate security attribute settings for the Portal. Bit 31 (Lsb) of this field must be 1 (enabled) for this field to contain significant information. If Bit 31 is enabled, this signifies that the iSNS server can be used to store and distribute security policies and settings for iSNS clients (i.e., iSCSI devices). Bit 30 must be 1 for bits 25-29 to contain significant information. All other bits are reserved for non-IKE/IPSec security mechanisms to be specified in the future.

Bit Position	Flag Description
25	1 = Tunnel Mode Preferred; 0 = No Preference
26	1 = Transport Mode Preferred; 0 = No Preference
27	1 = Perfect Forward Secrecy (PFS) Enabled; 0 = PFS Disabled
28	1 = Aggressive Mode Enabled; 0 = Disabled
29	1 = Main Mode Enabled; 0 = MM Disabled
30	1 = IKE/IPSec Enabled; 0 = IKE/IPSec Disabled
31 (Lsb)	1 = Bitmap VALID; 0 = INVALID
All others	RESERVED

6.3.10. Portal ISAKMP Phase-1 Proposals

This field contains the IKE Phase-1 proposal listing in decreasing order of preference of the protection suites acceptable to protect all IKE Phase-2 messages sent and received by the Portal. This includes Phase-2 SAs from the iSNS client to the iSNS server as well as to peer iFCP and/or iSCSI devices. This attribute contains the SA payload, proposal payload(s), and transform payload(s) in the ISAKMP format defined in [RFC2408].

This field should be used if the implementer wishes to define phase-1 SA security configuration on a per-Portal basis, as opposed to on a per-Network Entity basis. If the implementer desires to have a single phase-1 SA security configuration to protect all phase-2 traffic regardless of the interface used, then the Entity Phase-1 Proposal (Section 6.2.9) should be used.

6.3.11. Portal ISAKMP Phase-2 Proposals

This field contains the IKE Phase-2 proposal, in ISAKMP format [RFC2408], listing in decreasing order of preference the security proposals acceptable to protect traffic sent and received by the Portal. This field is used only if bits 31, 30, and 29 of the

Security Bitmap (see 6.3.9) are enabled. This attribute contains the SA payload, proposal payload(s), and associated transform payload(s) in the ISAKMP format defined in [RFC2408].

6.3.12. Portal Certificate

This attribute contains one or more X.509 certificates that are a credential of the Portal. This certificate is used to identify and authenticate communications to the IP address and TCP/UDP Port supported by the Portal. The format of the X.509 certificate is specified in [RFC3280]. This certificate MUST contain a Subject Name with an empty sequence and MUST contain a SubjectAltName extension encoded with the iPAddress type. The Portal IP Address (Section 6.3.1) of the identified Portal SHALL be stored in the SubjectAltName field of the certificate.

6.4. iSCSI Node-Keyed Attributes

The following attributes are stored in the iSNS database using the iSCSI Name attribute as the key. Each set of Node-Keyed attributes is associated with one Entity Identifier object key.

Although the iSCSI Name key is associated with one Entity Identifier, it is unique across the entire iSNS database.

6.4.1. iSCSI Name

This is a variable-length UTF-8 encoded NULL-terminated text-based description of up to 224 bytes. This key attribute is required for iSCSI Storage Nodes and is provided by the iSNS client. The registered iSCSI Name MUST conform to the format described in [iSCSI] for iSCSI Names. The maximum size for an iSCSI Name is 223 bytes. Including the NULL character and 4-byte alignment (see Section 5.3.1), the maximum iSCSI Name field size is 224 bytes.

If an iSCSI Name is registered without an EID key, then a Network Entity SHALL be created and an EID assigned. The assigned EID SHALL be returned in the registration response as an operating attribute.

This field MUST be normalized according to the stringprep template [STRINGPREP] before it is stored in the iSNS database.

6.4.2. iSCSI Node Type

This required 32-bit field is a bitmap indicating the type of iSCSI Storage Node. The bit positions are defined below. A set bit (1) indicates that the Node has the corresponding characteristics.

Bit Position	Node Type
29	Control
30	Initiator
31 (Lsb)	Target
All others	RESERVED

If the Target bit is set to 1, then the Node represents an iSCSI target. The Target bit MAY be set by iSNS clients using the iSNSP.

If the Initiator bit is set to 1, then the Node represents an iSCSI initiator. The Initiator bit MAY be set by iSNS clients using the iSNSP.

If the control bit is set to 1, then the Node represents a gateway, a management station, a backup iSNS server, or another device that is not an initiator or target, but that requires the ability to send and receive iSNSP messages, including state change notifications. Setting the control bit is an administrative task that MUST be performed on the iSNS server; iSNS clients SHALL NOT be allowed to change this bit using the iSNSP.

This field MAY be used by the iSNS server to distinguish among permissions by different iSCSI Node types for accessing various iSNS functions. More than one Node Type bit may be simultaneously enabled.

6.4.3. iSCSI Node Alias

This is a variable-length UTF-8 encoded NULL-terminated text-based description of up to 256 bytes. The Alias is a user-readable description of the Node entry in the iSNS database.

6.4.4. iSCSI Node SCN Bitmap

The iSCSI Node SCN Bitmap indicates events for which the registering iSNS client wishes to receive a notification message. The following table displays events that result in notifications, and the bit field in the SCN Bitmap that, when enabled, results in the corresponding notification.

Note that this field is of dual use: it is used in the SCN registration process to define interested events that will trigger an SCN message, and it is also contained in each SCN message itself, to indicate the type of event that triggered the SCN message. A set bit (1) indicates the corresponding type of SCN.

Bit Position	Flag Description
24	INITIATOR AND SELF INFORMATION ONLY
25	TARGET AND SELF INFORMATION ONLY
26	MANAGEMENT REGISTRATION/SCN
27	OBJECT REMOVED
28	OBJECT ADDED
29	OBJECT UPDATED
30	DD/DDS MEMBER REMOVED (Mgmt Reg/SCN only)
31 (Lsb)	DD/DDS MEMBER ADDED (Mgmt Reg/SCN only)
All others	RESERVED

DD/DDS MEMBER REMOVED indicates that an existing member of a Discovery Domain and/or Discovery Domain Set has been removed.

DD/DDS MEMBER ADDED indicates that a new member was added to an existing DD and/or DDS.

OBJECT REMOVED, OBJECT ADDED, and OBJECT UPDATED indicate a Network Entity, Portal, Storage Node, FC Device, DD, and/or DDS object was removed from, added to, or updated in the Discovery Domain or in the iSNS database (Control Nodes only).

Regular SCNs provide information about objects that are updated in, added to or removed from Discovery Domains of which the Storage Node is a member. An SCN or SCN registration is considered a regular SCN or regular SCN registration if the MANAGEMENT REGISTRATION/SCN flag is cleared. All iSNS clients may register for regular SCNs.

Management SCNs provide information about all changes to the network, regardless of discovery domain membership. Registration for management SCNs is indicated by setting bit 26 to 1. Only Control Nodes may register for management SCNs. Bits 30 and 31 may only be enabled if bit 26 is set to 1.

TARGET AND SELF INFORMATION ONLY SCNs (bit 25) provides information only about changes to target devices, or if the iSCSI Storage Node itself has undergone a change. Similarly, INITIATOR AND SELF INFORMATION ONLY SCNs (bit 24) provides information only about changes to initiator Nodes, or to the target itself.

6.4.5. iSCSI Node Index

The iSCSI Node Index is a 4-byte non-zero integer value used as a key that uniquely identifies each iSCSI Storage Node registered in the iSNS database. Upon initial registration of the iSCSI Storage Node, the iSNS server assigns an unused value for the iSCSI Node Index. Each iSCSI Node MUST be assigned a value for the iSCSI Node Index that is not assigned to any other iSCSI Storage Node. Furthermore, iSCSI Node Index values for recently deregistered iSCSI Storage Nodes SHOULD NOT be reused in the short term.

The iSCSI Node Index may be used as a key to represent a registered Node in situations where the iSCSI Name is too long to be used as a key. An example of this is when SNMP is used for management, as described in Section 2.10.

The value assigned for the iSCSI Node Index SHALL persist as long as the iSCSI Storage Node is registered in the iSNS database or a member of a Discovery Domain. An iSCSI Node Index value that is assigned for a Storage Node SHALL NOT be used for any other Storage Node as long as the original node is registered in the iSNS database or a member of a Discovery Domain.

6.4.6. WWNN Token

This field contains a globally unique 64-bit integer value that can be used to represent the World Wide Node Name of the iSCSI device in a Fibre Channel fabric. This identifier is used during the device registration process and MUST conform to the requirements in [FC-FS].

The FC-iSCSI gateway uses the value found in this field to register the iSCSI device in the Fibre Channel name server. It is stored in the iSNS server to prevent conflict when "proxy" WWNN values are assigned to iSCSI initiators establishing storage sessions to devices in the FC fabric.

If the iSNS client does not assign a value for WWNN Token, then the iSNS server SHALL provide a value for this field upon initial registration of the iSCSI Storage Node. The process by which the WWNN Token is assigned by the iSNS server MUST conform to the following requirements:

1. The assigned WWNN Token value MUST be unique among all WWN entries in the existing iSNS database, and among all devices that can potentially be registered in the iSNS database.

2. Once the value is assigned, the iSNS server MUST persistently save the mapping between the WWNN Token value and registered iSCSI Name. That is, successive re-registrations of the iSCSI Storage Node keyed by the same iSCSI Name maintain the original mapping to the associated WWNN Token value in the iSNS server. Similarly, the mapping SHALL be persistent across iSNS server reboots. Once assigned, the mapping can only be changed if a DevAttrReg message from an authorized iSNS client explicitly provides a different WWNN Token value.

3. Once a WWNN Token value has been assigned and mapped to an iSCSI name, that WWNN Token value SHALL NOT be reused or mapped to any other iSCSI name.

4. The assigned WWNN Token value MUST conform to the formatting requirements of [FC-FS] for World Wide Names (WWNs).

An iSNS client, such as an FC-iSCSI gateway or the iSCSI initiator, MAY register its own WWNN Token value or overwrite the iSNS Server-supplied WWNN Token value, if it wishes to supply its own iSCSI-FC name mapping. This is accomplished using the DevAttrReg message with the WWNN Token (tag=37) as an operating attribute. Once overwritten, the new WWNN Token value MUST be stored and saved by the iSNS server, and all requirements specified above continue to apply. If an iSNS client attempts to register a value for this field that is not unique in the iSNS database or that is otherwise invalid, then the registration SHALL be rejected with an Status Code of 3 (Invalid Registration).

There MAY be matching records in the iSNS database for the Fibre Channel device specified by the WWNN Token. These records may contain device attributes for that FC device registered in the Fibre Channel fabric name server.

6.4.7. iSCSI Node Next Index

This is a virtual attribute containing a 4-byte integer value that indicates the next available (i.e., unused) iSCSI Node Index value. This attribute may only be queried; the iSNS server SHALL return an error code of 3 (Invalid Registration) to any client that attempts to register a value for this attribute. A Message Key is not required when exclusively querying for this attribute.

The iSCSI Node Next Index MAY be used by an SNMP client to create an entry in the iSNS server. SNMP requirements are described in Section 2.10.

6.4.8. iSCSI AuthMethod

This attribute contains a NULL-terminated string of UTF-8 text listing the iSCSI authentication methods enabled for this iSCSI Storage Node, in order of preference. The text values used to identify iSCSI authentication methods are embedded in this string attribute and delineated by a comma. The text values are identical to those found in the main iSCSI document [iSCSI]; additional vendor-specific text values are also possible.

Text Value	Description	Reference
KB5	Kerberos V5	[RFC1510]
SPKM1	Simple Public Key GSS-API	[RFC2025]
SPKM2	Simple Public Key GSS-API	[RFC2025]
SRP	Secure Remote Password	[RFC2945]
CHAP	Challenge Handshake Protocol	[RFC1994]
none	No iSCSI Authentication	

6.5. Portal Group (PG) Object-Keyed Attributes

The following attributes are used to associate Portal and iSCSI Storage Node objects. PG objects are stored in the iSNS database using the PG iSCSI Name, the PG Portal IP Address, and the PG Portal TCP/UDP Port as keys. New PG objects are implicitly or explicitly created at the time that the corresponding Portal and/or iSCSI Storage Node objects are registered. Section 3.4 has a general discussion of PG usage. For further details on use of Portal Groups, see [iSCSI].

6.5.1. Portal Group iSCSI Name

This is the iSCSI Name for the iSCSI Storage Node that is associated with the PG object. This name MAY represent an iSCSI Storage Node not currently registered in the server.

6.5.2. PG Portal IP Addr

This is the Portal IP Address attribute for the Portal that is associated with the PG object. This Portal IP Address MAY be that of a Portal that is not currently registered in the server.

6.5.3. PG Portal TCP/UDP Port

This is the Portal TCP/UDP Port attribute for the Portal that is associated with the PG object. This Portal TCP/UDP Port MAY be that of a Portal that is not currently registered in the server.

6.5.4. Portal Group Tag (PGT)

This field is used to group Portals in order to coordinate connections in a session across Portals to a specified iSCSI Node. The PGT is a value in the range of 0-65535, or NULL. A NULL PGT value is registered by using 0 for the length in the TLV during registration. The two least significant bytes of the value contain the PGT for the object. The two most significant bytes are reserved. If a PGT value is not explicitly registered for an iSCSI Storage Node and Portal pair, then the PGT value SHALL be implicitly registered as 0x00000001.

6.5.5. Portal Group Index

The PG Index is a 4-byte non-zero integer value used as a key that uniquely identifies each PG object registered in the iSNS database. Upon initial registration of a PG object, the iSNS server MUST assign an unused value for the PG Index. Furthermore, PG Index values for recently deregistered PG objects SHOULD NOT be reused in the short term.

The PG Index MAY be used as the key to reference a registered PG in situations where a unique index for each PG object is required. It MAY also be used as the message key in an iSNS message to query or update a pre-existing PG object. An example of this is when SNMP is used for management, as described in Section 2.10. The value assigned for the PG Index SHALL persist as long as the server is active.

6.5.6. Portal Group Next Index

The PG Next Index is a virtual attribute containing a 4-byte integer value that indicates the next available (i.e., unused) PG Index value. This attribute may only be queried; the iSNS server SHALL return an error code of 3 (Invalid Registration) to any client that attempts to register a value for this attribute. A Message Key is not required when exclusively querying for this attribute.

The Portal Group Next Index MAY be used by an SNMP client to create an entry in the iSNS server. SNMP requirements are described in Section 2.10.

6.6. FC Port Name-Keyed Attributes

The following attributes are registered in the iSNS database using the FC Port World Wide Name (WWPN) attribute as the key. Each set of FC Port-Keyed attributes is associated with one Entity Identifier object key.

Although the FC Port World Wide Name is associated with one Entity Identifier, it is also globally unique.

6.6.1. FC Port Name (WWPN)

This 64-bit identifier uniquely defines the FC Port, and it is the World Wide Port Name (WWPN) of the corresponding Fibre Channel device. This attribute is the key for the iFCP Storage Node. This globally unique identifier is used during the device registration process, and it uses a value conforming to IEEE EUI-64 [EUI-64].

6.6.2. Port ID (FC_ID)

The Port Identifier is a Fibre Channel address identifier assigned to an N_Port or NL_Port during fabric login. The format of the Port Identifier is defined in [FC-FS]. The least significant 3 bytes contain this address identifier. The most significant byte is RESERVED.

6.6.3. FC Port Type

Indicates the type of FC port. Encoded values for this field are listed in the following table:

Type	Description
0x0000	Unidentified/Null Entry
0x0001	Fibre Channel N_Port
0x0002	Fibre Channel NL_Port
0x0003	Fibre Channel F/NL_Port
0x0004-0080	RESERVED
0x0081	Fibre Channel F_Port
0x0082	Fibre Channel FL_Port
0x0083	RESERVED
0x0084	Fibre Channel E_Port
0x0085-00FF	RESERVED
0xFF11	RESERVED
0xFF12	iFCP Port
0xFF13-FFFF	RESERVED

6.6.4. Symbolic Port Name

This is a variable-length UTF-8 encoded NULL-terminated text-based description of up to 256 bytes that is associated with the iSNS-registered FC Port Name in the network.

6.6.5. Fabric Port Name (FWWN)

This 64-bit identifier uniquely defines the fabric port. If the port of the FC Device is attached to a Fibre Channel fabric port with a registered Port Name, then that fabric Port Name SHALL be indicated in this field.

6.6.6. Hard Address

This field is the requested hard address 24-bit NL Port Identifier, included in the iSNSP for compatibility with Fibre Channel Arbitrated Loop devices and topologies. The least significant 3 bytes of this field contain the address. The most significant byte is RESERVED.

6.6.7. Port IP Address

The Fibre Channel IP address associated with the FC Port. When this field contains an IPv4 value, it is stored as an IPv4-mapped IPv6 address. That is, the most significant 10 bytes are set to 0x00, with the next two bytes set to 0xFFFF [RFC2373]. When an IPv6 value is contained in this field, then the entire 16-byte field is used.

6.6.8. Class of Service (COS)

This 32-bit bit-map field indicates the Fibre Channel Class of Service types that are supported by the registered port. In the following table, a set bit (1) indicates a Class of Service supported.

Bit Position	Description
29	Fibre Channel Class 2 Supported
28	Fibre Channel Class 3 Supported

6.6.9. FC-4 Types

This 32-byte field indicates the FC-4 protocol types supported by the associated port. This field can be used to support Fibre Channel devices and is consistent with FC-GS-4.

6.6.10. FC-4 Descriptor

This is a variable-length UTF-8 encoded NULL-terminated text-based description of up to 256 bytes that is associated with the iSNS-registered device port in the network. This field can be used to support Fibre Channel devices and is consistent with FC-GS-4.

6.6.11. FC-4 Features

This is a 128-byte array, 4 bits per type, for the FC-4 protocol types supported by the associated port. This field can be used to support Fibre Channel devices and is consistent with FC-GS-4.

6.6.12. iFCP SCN Bitmap

This field indicates the events the iSNS client is interested in. These events can cause SCNs to be generated. SCNs provide information about objects that are updated in, added to or removed from Discovery Domains of which the source and destination are a member. Management SCNs provide information about all changes to the network. A set bit (1) indicates the type of SCN for the bitmap as follows:

Bit Position	Flag Description
24	INITIATOR AND SELF INFORMATION ONLY
25	TARGET AND SELF INFORMATION ONLY
26	MANAGEMENT REGISTRATION/SCN
27	OBJECT REMOVED
28	OBJECT ADDED
29	OBJECT UPDATED
30	DD/DDS MEMBER REMOVED (Mgmt Reg/SCN only)
31 (Lsb)	DD/DDS MEMBER ADDED (Mgmt Reg/SCN only)
All others	RESERVED

Further information on the use of the bit positions specified above can be found in Section 6.4.4.

6.6.13. Port Role

This required 32-bit field is a bitmap indicating the type of iFCP Storage Node. The bit fields are defined below. A set bit indicates the Node has the corresponding characteristics.

Bit Position	Node Type
29	Control
30	FCP Initiator
31 (Lsb)	FCP Target
All Others	RESERVED

If the 'Target' bit is set to 1, then the port represents an FC target. Setting of the 'Target' bit MAY be performed by iSNS clients using the iSNSP.

If the 'Initiator' bit is set to 1, then the port represents an FC initiator. Setting of the 'Initiator' bit MAY be performed by iSNS clients using the iSNSP.

If the 'Control' bit is set to 1, then the port represents a gateway, a management station, an iSNS backup server, or another device.

This is usually a special device that is neither an initiator nor a target, which requires the ability to send and receive iSNSP messages, including state-change notifications. Setting the control bit is an administrative task that MUST be administratively configured on the iSNS server; iSNS clients SHALL NOT be allowed to change this bit using the iSNSP.

This field MAY be used by the iSNS server to distinguish among permissions by different iSNS clients. For example, an iSNS server implementation may be administratively configured to allow only targets to receive ESIs, or to permit only Control Nodes to add, modify, or delete discovery domains.

6.6.14. Permanent Port Name (PPN)

The Permanent Port Name can be used to support Fibre Channel devices and is consistent with the PPN description in FC-GS-4 [FC-GS-4]. The format of the PPN is identical to the FC Port Name WWPN attribute format.

6.7. Node-Keyed Attributes

The following attributes are registered in the iSNS database using the FC Node Name (WWNN) attribute as the key. Each set of FC Node-Keyed attributes represents a single device and can be associated with many FC Ports.

The FC Node Name is unique across the entire iSNS database.

6.7.1. FC Node Name (WWNN)

The FC Node Name is a 64-bit identifier that is the World Wide Node Name (WWNN) of the corresponding Fibre Channel device. This attribute is the key for the FC Device. This globally unique identifier is used during the device registration process, and it uses a value conforming to IEEE EUI-64 [EUI-64].

6.7.2. Symbolic Node Name

This is a variable-length UTF-8 encoded NULL-terminated text-based description of up to 256 bytes that is associated with the iSNS-registered FC Device in the network.

6.7.3. Node IP Address

This IP address is associated with the device Node in the network. This field is included for compatibility with Fibre Channel. When this field contains an IPv4 value, it is stored as an IPv4-mapped IPv6 address. That is, the most significant 10 bytes are set to 0x00, with the next two bytes set to 0xFFFF [RFC2373]. When an IPv6 value is contained in this field, the entire 16-byte field is used.

6.7.4. Node IPA

This field is the 8-byte Fibre Channel Initial Process Associator (IPA) associated with the device Node in the network. The initial process associator is used for communication between Fibre Channel devices.

6.7.5. Proxy iSCSI Name

This is a variable-length UTF-8 encoded NULL-terminated text-based field that contains the iSCSI Name used to represent the FC Node in the IP network. It is used as a pointer to the matching iSCSI Name entry in the iSNS server. Its value is usually registered by an FC-iSCSI gateway connecting the IP network to the fabric containing the FC device.

Note that if this field is used, there SHOULD be a matching entry in the iSNS database for the iSCSI device specified by the iSCSI name. The database entry should include the full range of iSCSI attributes needed for discovery and management of the "iSCSI proxy image" of the FC device.

6.8. Other Attributes

The following are not attributes of the previously-defined objects.

6.8.1. FC-4 Type Code

This is a 4-byte field used to provide a FC-4 type during a FC-4 Type query. The FC-4 types are consistent with the FC-4 Types as defined in FC-FS. Byte 0 contains the FC-4 type. All other bytes are reserved.

6.8.2. iFCP Switch Name

The iFCP Switch Name is a 64-bit World Wide Name (WWN) identifier that uniquely identifies a distinct iFCP gateway in the network. This globally unique identifier is used during the switch registration/FC_DOMAIN_ID assignment process. The iFCP Switch Name value used MUST conform to the requirements stated in [FC-FS] for World Wide Names. The iSNS server SHALL track the state of all FC_DOMAIN_ID values that have been allocated to each iFCP Switch Name. If a given iFCP Switch Name is deregistered from the iSNS database, then all FC_DOMAIN_ID values allocated to that iFCP Switch Name SHALL be returned to the unused pool of values.

6.8.3. iFCP Transparent Mode Commands

6.8.3.1. Preferred ID

This is a 4-byte unsigned integer field, and it is the requested value that the iSNS client wishes to use for the FC_DOMAIN_ID. The iSNS server SHALL grant the iSNS client the use of the requested value as the FC_DOMAIN_ID, if the requested value has not already been allocated. If the requested value is not available, the iSNS server SHALL return a different value that has not been allocated.

6.8.3.2. Assigned ID

This is a 4-byte unsigned integer field that is used by an iFCP gateway to reserve its own unique FC_DOMAIN_ID value from the range 1 to 239. When a FC_DOMAIN_ID is no longer required, it SHALL be released by the iFCP gateway using the RlseDomId message. The iSNS server MUST use the Entity Status Inquiry message to determine whether an iFCP gateway is still present on the network.

6.8.3.3. Virtual_Fabric_ID

This is a variable-length UTF-8 encoded NULL-terminated text-based field of up to 256 bytes. The Virtual_Fabric_ID string is used as a key attribute to identify a range of non-overlapping FC_DOMAIN_ID values to be allocated using RqstDomId. Each Virtual_Fabric_ID string submitted by an iSNS client SHALL have its own range of non-overlapping FC_DOMAIN_ID values to be allocated to iSNS clients.

6.9. iSNS Server-Specific Attributes

Access to the following attributes may be administratively controlled. These attributes are specific to the iSNS server instance; the same value is returned for all iSNS clients accessing the iSNS server. Only query messages may be performed on these attributes. Attempted registrations of values for these attributes SHALL return a status code of 3 (Invalid Registration).

A query for an iSNS Server-Specific attribute MUST contain the identifying key attribute (i.e., iSCSI Name or FC Port Name WWPN) of the Node originating the registration or query message as the Source and Message Key attributes. The Operating Attributes are the server-specific attributes being registered or queried.

6.9.1. iSNS Server Vendor OUI

This attribute is the OUI (Organizationally Unique Identifier) [802-1990] identifying the specific vendor implementing the iSNS server. This attribute can only be queried; iSNS clients SHALL NOT be allowed to register a value for the iSNS Server Vendor OUI.

6.10. Vendor-Specific Attributes

iSNS server implementations MAY define vendor-specific attributes for private use. These attributes MAY be used to store optional data that is registered and/or queried by iSNS clients in order to gain optional capabilities. Note that any implementation of vendor-specific attributes in the iSNS server SHALL NOT impose any form of mandatory behavior on the part of the iSNS client.

The tag values used for vendor-specific and user-specific use are defined in Section 6.1. To avoid misinterpreting proprietary attributes, the vendor's own OUI (Organizationally Unique Identifier) MUST be placed in the upper three bytes of the attribute value field itself.

The OUI is defined in IEEE Std 802-1990 and is the same constant used to generate 48 bit Universal LAN MAC addresses. A vendor's own iSNS implementation will then be able to recognize the OUI in the attribute field and be able to execute vendor-specific handling of the attribute.

6.10.1. Vendor-Specific Server Attributes

Attributes with tags in the range 257 to 384 are vendor-specific or site-specific attributes of the iSNS server. Values for these attributes are administratively set by the specific vendor providing the iSNS server implementation. Query access to these attributes may be administratively controlled. These attributes are unique for each logical iSNS server instance. Query messages for these attributes SHALL use the key identifier (i.e., iSCSI Name or FC Port Name WWPN) for both the Source attribute and Message Key attribute. These attributes can only be queried; iSNS clients SHALL NOT be allowed to register a value for server attributes.

6.10.2. Vendor-Specific Entity Attributes

Attributes in the range 385 to 512 are vendor-specific or site-specific attributes used to describe the Network Entity object. These attributes are keyed by the Entity Identifier attribute (tag=1).

6.10.3. Vendor-Specific Portal Attributes

Attributes in the range 513 to 640 are vendor-specific or site-specific attributes used to describe the Portal object. These attributes are keyed by the Portal IP-Address (tag=16) and Portal TCP/UDP Port (tag=17).

6.10.4. Vendor-Specific iSCSI Node Attributes

Attributes in the range 641 to 768 are vendor-specific or site-specific attributes used to describe the iSCSI Node object. These attributes are keyed by the iSCSI Name (tag=32).

6.10.5. Vendor-Specific FC Port Name Attributes

Attributes in the range 769 to 896 are vendor-specific or site-specific attributes used to describe the N_Port Port Name object. These attributes are keyed by the FC Port Name WWPN (tag=64).

6.10.6. Vendor-Specific FC Node Name Attributes

Attributes in the range 897 to 1024 are vendor-specific or site-specific attributes used to describe the FC Node Name object. These attributes are keyed by the FC Node Name WWNN (tag=96).

6.10.7. Vendor-Specific Discovery Domain Attributes

Attributes in the range 1025 to 1280 are vendor-specific or site-specific attributes used to describe the Discovery Domain object. These attributes are keyed by the DD_ID (tag=104).

6.10.8. Vendor-Specific Discovery Domain Set Attributes

Attributes in the range 1281 to 1536 are vendor-specific or site-specific attributes used to describe the Discovery Domain Set object. These attributes are keyed by the DD Set ID (tag=101)

6.10.9. Other Vendor-Specific Attributes

Attributes in the range 1537 to 2048 can be used for key and non-key attributes that describe new vendor-specific objects specific to the vendor’s iSNS server implementation.

6.11. Discovery Domain Registration Attributes

6.11.1. DD Set ID Keyed Attributes

6.11.1.1. Discovery Domain Set ID (DDS ID)

The DDS ID is an unsigned non-zero integer identifier used in the iSNS directory database as a key to indicate a Discovery Domain Set uniquely. A DDS is a collection of Discovery Domains that can be enabled or disabled by a management station. This value is used as a key for DDS attribute queries. When a Discovery Domain is registered, it is initially not in any DDS.

If the iSNS client does not provide a DDS_ID in a DDS registration request message, the iSNS server SHALL generate a DDS_ID value that is unique within the iSNS database for that new DDS. The created DDS ID SHALL be returned in the response message. The DDS ID value of 0 is reserved, and the DDS ID value of 1 is used for the default DDS (see Section 2.2.2).

6.11.1.2. Discovery Domain Set Symbolic Name

A variable-length UTF-8 encoded NULL-terminated text-based field of up to 256 bytes. This is a user-readable field used to assist a network administrator in tracking the DDS function. When a client registers a DDS symbolic name, the iSNS server SHALL verify it is unique. If the name is not unique, then the DDS registration SHALL be rejected with an "Invalid Registration" Status Code. The invalid attribute(s), in this case the DDS symbolic name, SHALL be included in the response.

6.11.1.3. Discovery Domain Set Status

The DDS_Status field is a 32-bit bitmap indicating the status of the DDS. Bit 0 of the bitmap indicates whether the DDS is Enabled (1) or Disabled (0). The default value for the DDS Enabled flag is Disabled (0).

Bit Position	DDS Status
31 (Lsb)	DDS Enabled (1) / DDS Disabled (0)
All others	RESERVED

6.11.1.4. Discovery Domain Set Next ID

This is a virtual attribute containing a 4-byte integer value that indicates the next available (i.e., unused) Discovery Domain Set Index value. This attribute may only be queried; the iSNS server

SHALL return an error code of 3 (Invalid Registration) to any client that attempts to register a value for this attribute. A Message Key is not required when exclusively querying for this attribute.

The Discovery Domain Set Next Index MAY be used by an SNMP client to create an entry in the iSNS server. SNMP requirements are described in Section 2.10.

6.11.2. DD ID Keyed Attributes

6.11.2.1. Discovery Domain ID (DD ID)

The DD ID is an unsigned non-zero integer identifier used in the iSNS directory database as a key to identify a Discovery Domain uniquely. This value is used as the key for any DD attribute query. If the iSNS client does not provide a DD_ID in a DD registration request message, the iSNS server SHALL generate a DD_ID value that is unique within the iSNS database for that new DD (i.e., the iSNS client will be registered in a new DD). The created DD ID SHALL be returned in the response message. The DD ID value of 0 is reserved, and the DD ID value of 1 is used for the default DD (see Section 2.2.2).

6.11.2.2. Discovery Domain Symbolic Name

A variable-length UTF-8 encoded NULL-terminated text-based field of up to 256 bytes. When a client registers a DD symbolic name, the iSNS server SHALL verify it is unique. If the name is not unique, then the DD registration SHALL be rejected with an "Invalid Registration" Status Code. The invalid attribute(s), in this case the DD symbolic name, SHALL be included in the response.

6.11.2.3. Discovery Domain Member: iSCSI Node Index

This is the iSCSI Node Index of a Storage Node that is a member of the DD. The DD may have a list of 0 to n members. The iSCSI Node Index is one alternative representation of membership in a Discovery Domain, the other alternative being the iSCSI Name. The Discovery Domain iSCSI Node Index is a 4-byte non-zero integer value.

The iSCSI Node Index can be used to represent a DD member in situations where the iSCSI Name is too long to be used. An example of this is when SNMP is used for management, as described in Section 2.10.

The iSCSI Node Index and the iSCSI Name stored as a member in a DD SHALL be consistent with the iSCSI Node Index and iSCSI Name attributes registered for the Storage Node object in the iSNS server.

6.11.2.4. Discovery Domain Member: iSCSI Name

A variable-length UTF-8 encoded NULL-terminated text-based field of up to 224 bytes. It indicates membership for the specified iSCSI Storage Node in the Discovery Domain. Note that the referenced Storage Node does not need to be actively registered in the iSNS database before the iSNS client uses this attribute. There is no limit to the number of members that may be in a DD. Membership is represented by the iSCSI Name of the iSCSI Storage Node.

6.11.2.5. Discovery Domain Member: FC Port Name

This 64-bit identifier attribute indicates membership for an iFCP Storage Node (FC Port) in the Discovery Domain. Note that the referenced Storage Node does not need to be actively registered in the iSNS database before the iSNS client uses this attribute. There is no limit to the number of members that may be in a DD. Membership is represented by the FC Port Name (WWPN) of the iFCP Storage Node.

6.11.2.6. Discovery Domain Member: Portal Index

This attribute indicates membership in the Discovery Domain for a Portal. It is an alternative representation for Portal membership to the Portal IP Address and Portal TCP/UDP Port. The referenced Portal MUST be actively registered in the iSNS database before the iSNS client uses this attribute.

6.11.2.7. Discovery Domain Member: Portal IP Address

This attribute and the Portal TCP/UDP Port attribute indicate membership in the Discovery Domain for the specified Portal. Note that the referenced Portal does not need to be actively registered in the iSNS database before the iSNS client uses this attribute.

6.11.2.8. Discovery Domain Member: Portal TCP/UDP Port

This attribute and the Portal IP Address attribute indicate membership in the Discovery Domain for the specified Portal. Note that the referenced Portal does not need to be actively registered in the iSNS database before the iSNS client uses this attribute.

6.11.2.9. Discovery Domain Features

The Discovery Domain Features is a bitmap indicating the features of this DD. The bit positions are defined below. A bit set to 1 indicates the DD has the corresponding characteristics.

Bit Position	DD Feature
31 (Lsb)	Boot List Enabled (1)/Boot List Disabled (0)
All others	RESERVED

Boot List: this feature indicates that the target(s) in this DD provides boot capabilities for the member initiators, as described in [iSCSI-boot].

6.11.2.10. Discovery Domain Next ID

This is a virtual attribute containing a 4-byte integer value that indicates the next available (i.e., unused) Discovery Domain Index value. This attribute may only be queried; the iSNS server SHALL return an error code of 3 (Invalid Registration) to any client that attempts to register a value for this attribute. A Message Key is not required when exclusively querying for this attribute.

7. Security Considerations

7.1. iSNS Security Threat Analysis

When the iSNS protocol is deployed, the interaction between iSNS server and iSNS clients is subject to the following security threats:

a) An attacker could alter iSNS protocol messages, such as to direct iSCSI and iFCP devices to establish connections with rogue peer devices, or to weaken/eliminate IPSec protection for iSCSI or iFCP traffic.

b) An attacker could masquerade as the real iSNS server using false iSNS heartbeat messages. This could cause iSCSI and iFCP devices to use rogue iSNS servers.

c) An attacker could gain knowledge about iSCSI and iFCP devices by snooping iSNS protocol messages. Such information could aid an attacker in mounting a direct attack on iSCSI and iFCP devices, such as a denial-of-service attack or outright physical theft.

To address these threats, the following capabilities are needed:

a) Unicast iSNS protocol messages may need to be authenticated. In addition, to protect against threat c), confidentiality support is desirable and is REQUIRED when certain functions of iSNS server are utilized.

b) Multicast iSNS protocol messages such as the iSNS heartbeat message may need to be authenticated. These messages need not be confidential since they do not leak critical information.

7.2. iSNS Security Implementation and Usage Requirements

If the iSNS server is used to distribute authorizations for communications between iFCP and iSCSI peer devices, IPsec ESP with null transform MUST be implemented, and non-null transform MAY be implemented. If a non-null transform is implemented, then the DES encryption algorithm SHOULD NOT be used.

If the iSNS server is used to distribute security policy for iFCP and iSCSI devices, then authentication, data integrity, and confidentiality MUST be supported and used. Where confidentiality is desired or required, IPSec ESP with non-null transform SHOULD be used, and the DES encryption algorithm SHOULD NOT be used.

If the iSNS server is used to provide the boot list for clients, as described in Section 6.11.2.9, then the iSCSI boot client SHOULD implement a secure iSNS connection.

In order to protect against an attacker masquerading as an iSNS server, client devices MUST support the ability to authenticate broadcast or multicast messages such as the iSNS heartbeat. The iSNS authentication block (which is identical in format to the SLP authentication block) SHALL be used for this purpose. iSNS clients MUST implement the iSNS authentication block and MUST support BSD value 0x002. If the iSNS server supports broadcast or multicast iSNS messages (i.e., the heartbeat), then the server MUST implement the iSNS authentication block and MUST support BSD value 0x002. Note that the authentication block is used only for iSNS broadcast or multicast messages and MUST NOT be used in unicast iSNS messages.

There is no requirement that the communicating identities in iSNS protocol messages be kept confidential. Specifically, the identity and location of the iSNS server is not considered confidential.

For protecting unicast iSNS protocol messages, iSNS servers supporting security MUST implement ESP in tunnel mode and MAY implement transport mode.

All iSNS implementations supporting security MUST support the replay protection mechanisms of IPsec.

iSNS security implementations MUST support both IKE Main Mode and Aggressive Mode for authentication, negotiation of security associations, and key management, using the IPSec DOI [RFC2407].

Manual keying SHOULD NOT be used since it does not provide the necessary rekeying support. Conforming iSNS security implementations MUST support authentication using a pre-shared key, and MAY support certificate-based peer authentication using digital signatures. Peer authentication using the public key encryption methods outlined in IKEs Sections 5.2 and 5.3 [RFC2409] SHOULD NOT be supported.

Conforming iSNS implementations MUST support both IKE Main Mode and Aggressive Mode. IKE Main Mode with pre-shared key authentication SHOULD NOT be used when either of the peers use dynamically assigned IP addresses. Although Main Mode with pre-shared key authentication offers good security in many cases, situations where dynamically assigned addresses are used force the use of a group pre-shared key, which is vulnerable to man-in-the-middle attack. IKE Identity Payload ID_KEY_ID MUST NOT be used.

When digital signatures are used for authentication, either IKE Main Mode or IKE Aggressive Mode MAY be used. In all cases, access to locally stored secret information (pre-shared key or private key for digital signing) MUST be suitably restricted, since compromise of the secret information nullifies the security properties of the IKE/IPsec protocols.

When digital signatures are used to achieve authentication, an IKE negotiator SHOULD use IKE Certificate Request Payload(s) to specify the certificate authority (or authorities) that are trusted in accordance with its local policy. IKE negotiators SHOULD check the pertinent Certificate Revocation List (CRL) before accepting a PKI certificate for use in IKE's authentication procedures.

When the iSNS server is used without security, IP block storage protocol implementations MUST support a negative cache for authentication failures. This allows implementations to avoid continually contacting discovered endpoints that fail authentication within IPsec or at the application layer (in the case of iSCSI Login). The negative cache need not be maintained within the IPsec implementation, but rather within the IP block storage protocol implementation.

7.3. Discovering Security Requirements of Peer Devices

Once communication between iSNS clients and the iSNS server has been secured through use of IPSec, the iSNS client devices have the capability to discover the security settings that they need to use for their peer-to-peer communications using the iSCSI and/or iFCP protocols. This provides a potential scaling advantage over device-by-device configuration of individual security policies for each iSCSI and iFCP device.

The iSNS server stores security settings for each iSCSI and iFCP device interface. These security settings, which can be retrieved by authorized hosts, include use or non-use of IPSec, IKE, Main Mode, and Aggressive Mode. For example, IKE may not be enabled for a particular interface of a peer device. If a peer device can learn of this in advance by consulting the iSNS server, it will not need to waste time and resources attempting to initiate an IKE phase 1 session with that peer device interface.

If iSNS is used for this purpose, then the minimum information that should be learned from the iSNS server is the use or non-use of IKE and IPSec by each iFCP or iSCSI peer device interface. This information is encoded in the Security Bitmap field of each Portal of the peer device, and is applicable on a per-interface basis for the peer device. iSNS queries for acquiring security configuration data about peer devices MUST be protected by IPSec/ESP authentication.

7.4. Configuring Security Policies of iFCP/iSCSI Devices

Use of iSNS for distribution of security policies offers the potential to reduce the burden of manual device configuration, and to decrease the probability of communications failures due to incompatible security policies. If iSNS is used to distribute security policies, then IPSec authentication, data integrity, and confidentiality MUST be used to protect all iSNS protocol messages.

The complete IKE/IPSec configuration of each iFCP and/or iSCSI device can be stored in the iSNS server, including policies that are used for IKE Phase 1 and Phase 2 negotiations between client devices. The IKE payload format includes a series of one or more proposals that the iSCSI or iFCP device will use when negotiating the appropriate IPsec policy to use to protect iSCSI or iFCP traffic.

In addition, the iSCSI Authentication Methods used by each iSCSI device can also be stored in the iSNS server. The iSCSI AuthMethod field (tag=42) contains a null-terminated string embedded with the text values indicating iSCSI authentication methods to be used by that iSCSI device.

Note that iSNS distribution of security policy is not necessary if the security settings can be determined by other means, such as manual configuration or IPsec security policy distribution. If a network entity has already obtained its security configuration via other mechanisms, then it MUST NOT request security policy via iSNS.

7.5. Resource Issues

The iSNS protocol is lightweight and will not generate a significant amount of traffic. iSNS traffic is characterized by occasional registration, notification, and update messages that do not consume significant amounts of bandwidth. Even software-based IPSec implementations should not have a problem handling the traffic loads generated by the iSNS protocol.

To fulfill iSNS security requirements, the only additional resources needed beyond what is already required for iSCSI and iFCP involve the iSNS server. Because iSCSI and iFCP end nodes are already required to implement IKE and IPSec, these existing requirements can also be used to fulfill IKE and IPSec requirements for iSNS clients.

7.6. iSNS Interaction with IKE and IPSec

When IPSec security is enabled, each iSNS client with at least one Storage Node that is registered in the iSNS database SHALL maintain at least one phase-1 security association with the iSNS server. All iSNS protocol messages between iSNS clients and the iSNS server SHALL be protected by a phase-2 security association.

When a Network Entity is removed from the iSNS database, the iSNS server SHALL send a phase-1 delete message to the associated iSNS client IKE peer, and tear down all phase-1 and phase-2 SAs associated with that iSNS client.

8. IANA Considerations

The well-known TCP and UDP port number for iSNS is 3205.

The standards action of this RFC creates two registries to be maintained by IANA in support of iSNSP and assigns initial values for both registries. The first registry is of Block Storage Protocols supported by iSNS. The second registry is a detailed registry of standard iSNS attributes that can be registered to and queried from the iSNS server. Note that this RFC uses the registry created for Block Structure Descriptor (BSD) in Section 15 of Service Location Protocol, Version 2 [RFC2608].

8.1. Registry of Block Storage Protocols

In order to maintain a registry of block storage protocols supported by iSNSP, IANA will assign a 32-bit unsigned integer number for each block storage protocol supported by iSNS. This number is stored in the iSNS database as the Entity Protocol. The initial set of values to be maintained by IANA for Entity Protocol is indicated in the

table in Section 6.2.2. Additional values for new block storage protocols to be supported by iSNS SHALL be assigned by the IPS WG Chairperson, or by a Designated Expert [RFC2434] appointed by the IETF Transport Area Director.

8.2. Registry of Standard iSNS Attributes

IANA is responsible for creating and maintaining the Registry of Standard iSNS Attributes. The initial list of iSNS attributes is described in Section 6. For each iSNS attribute this information MUST include, its tag value, the attribute length, and the tag values for the set of permissible registration and query keys that can be used for that attribute. The initial list of iSNS attributes to be maintained by IANA is indicated in Section 6.1.

Additions of new standard attributes to the Registry of Standard iSNS Attributes SHALL require IETF Consensus [RFC2434]. The RFC required for this process SHALL specify use of tag values reserved for IANA allocation in Section 6.1. The RFC SHALL specify as a minimum, the new attribute tag value, attribute length, and the set of permissible registration and query keys that can be used for the new attribute. The RFC SHALL also include a discussion of the reasons for the new attribute(s) and how the new attribute(s) are to be used.

As part of the process of obtaining IETF Consensus, the proposed RFC and its supporting documentation SHALL be made available to the IPS WG mailing list or, if the IPS WG is disbanded at the time, to a mailing list designated by the IETF Transport Area Director. The review and comment period SHALL last at least three months before the IPS WG Chair or a person designated by the IETF Transport Area Director decides either to reject the proposal or to forward the draft to the IESG for publication as an RFC. When the specification is published as an RFC, then IANA will register the new iSNS attribute(s) and make the registration available to the community.

8.3. Block Structure Descriptor (BSD) Registry

Note that IANA is already responsible for assigning and maintaining values used for the Block Structure Descriptor for the iSNS Authentication Block (see Section 5.5). Section 15 of [RFC2608] describes the process for allocation of new BSD values.

9. Normative References

[iSCSI] Satran, J., Meth, K., Sapuntzakis, C., Chadalapaka, M., and E. Zeidner, "Internet Small Computer Systems Interface (iSCSI)", RFC 3720, April 2004.

[iFCP] Monia, C., Mullendore, R., Travostino, F., Jeong, W., and M. Edwards, "iFCP - A Protocol for Internet Fibre Channel Storage Networking", RFC 4172, September 2005.

[iSNSOption] Monia, C., Tseng, J., and K. Gibbons, The IPv4 Dynamic Host Configuration Protocol (DHCP) Option for the Internet Storage Name Service, RFC 4174, September 2005.

[RFC2608] Guttman, E., Perkins, C., Veizades, J., and M. Day, "Service Location Protocol, Version 2 ", RFC 2608, June 1999.

[iSCSI-SLP] Bakke, M., Hufferd, J., Voruganti, K., Krueger, M., and T. Sperry, "Finding Internet Small Computer Systems Interface (iSCSI) Targets and Name Servers by Using Service Location Protocol version 2 (SLP), RFC 4018, April 2005.

[iSCSI-boot] Sarkar, P., Missimer, D., and C. Sapuntzakis, "Bootstrapping Clients using the Internet Samll Computer System Interface (iSCSI) Protocol", RFC 4173, September 2005.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[STRINGPREP] Bakke, M., "String Profile for Internet Small Computer Systems Interface (iSCSI) Names", RFC 3722, April 2004.

[NAMEPREP] Hoffman, P. Nameprep: A Stringprep Profile for Internationalized Domain Names, July 2002.

[RFC2407] Piper, D., "The Internet IP Security Domain of Interpretation for ISAKMP", RFC 2407, November 1998.

[RFC2408] Maughan, D., Schertler, M., Schneider, M., and J. Turner, "Internet Security Association and Key Management Protocol (ISAKMP)", RFC 2408, November 1998.

[RFC2409] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.

[EUI-64] Guidelines for 64-bit Global Identifier (EUI-64) Registration Authority, May 2001, IEEE

[RFC3279] Bassham, L., Polk, W., and R. Housley, "Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3279, April 2002.

[RFC3280] Housley, R., Polk, W., Ford, W., and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, April 2002.

[802-1990] IEEE Standards for Local and Metropolitan Area Networks: Overview and Architecture, Technical Committee on Computer Communications of the IEEE Computer Society, May 31, 1990

[FC-FS] Fibre Channel Framing and Signaling Interface, NCITS Working Draft Project 1331-D

10. Informative References

[iSNSMIB] Gibbons, K., et al., "Definitions of Managed Objects for iSNS (Internet Storage name Service)", Work in Progress, July 2003.

[X.509] ITU-T Recommendation X.509 (1997 E): Information Technology - Open Systems Interconnection - The Directory: Authentication Framework, June 1997

[FC-GS-4] Fibre Channel Generic Services-4 (work in progress), NCITS Working Draft Project 1505-D

[RFC1510] Kohl, J. and C. Neuman, "The Kerberos Network Authentication Service (V5)", RFC 1510, September 1993.

[RFC2025] Adams, C., "The Simple Public-Key GSS-API Mechanism (SPKM)", RFC 2025, October 1996.

[RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

[RFC2945] Wu, T., "The SRP Authentication and Key Exchange System", RFC 2945, September 2000.

[RFC1994] Simpson, W., "PPP Challenge Handshake Authentication Protocol (CHAP)", RFC 1994, August 1996.

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

[RFC3411] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", STD 62, RFC 3411, December 2002.

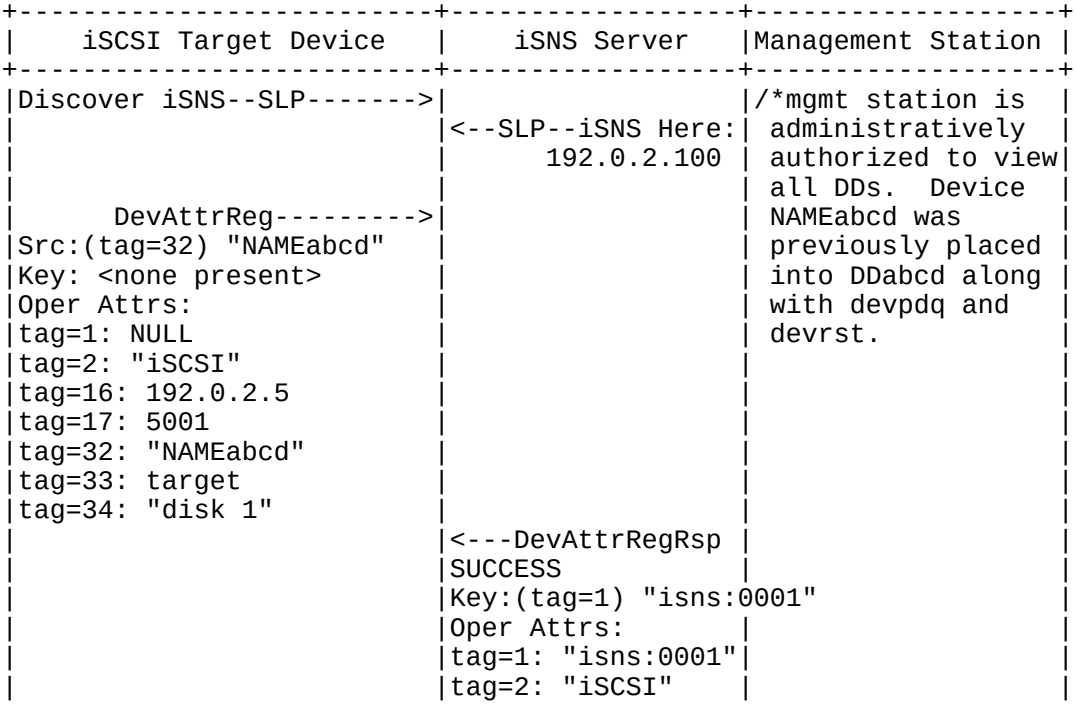
Appendix A: iSNS Examples

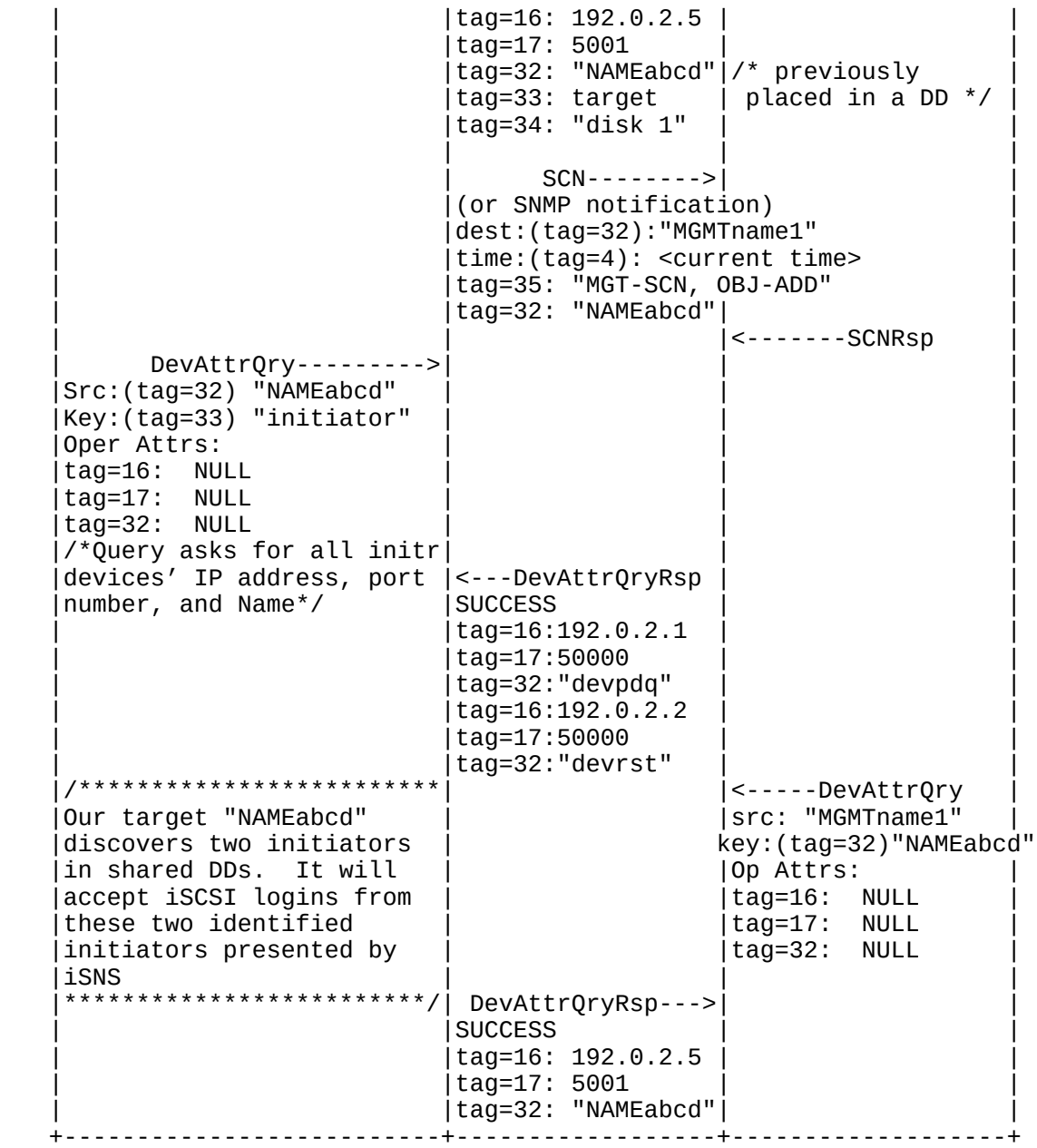
A.1. iSCSI Initialization Example

This example assumes an SLP Service Agent (SA) has been implemented on the iSNS host, and an SLP User Agent (UA) has been implemented on the iSNS initiator. See [RFC2608] for further details on SAs and UAs. This example also assumes that the target is configured to use the iSNS server, and have its access control policy subordinated to the iSNS server.

A.1.1. Simple iSCSI Target Registration

In this example, a simple target with a single iSCSI name registers with the iSNS server. The target is represented in the iSNS by an Entity containing one Storage Node, one Portal, and an implicitly registered Portal Group that provides a relationship between the Storage Node and Portal. The target has not been assigned a Fully Qualified Domain Name (FQDN) by the administrator. In this example, because a PG object is not explicitly registered, a Portal Group with a PGT of 1 is implicitly registered. In this example SLP is used to discover the location of the iSNS Server. An alternative is to use the iSNS DHCP option [iSNSOption] to discover the iSNS server.

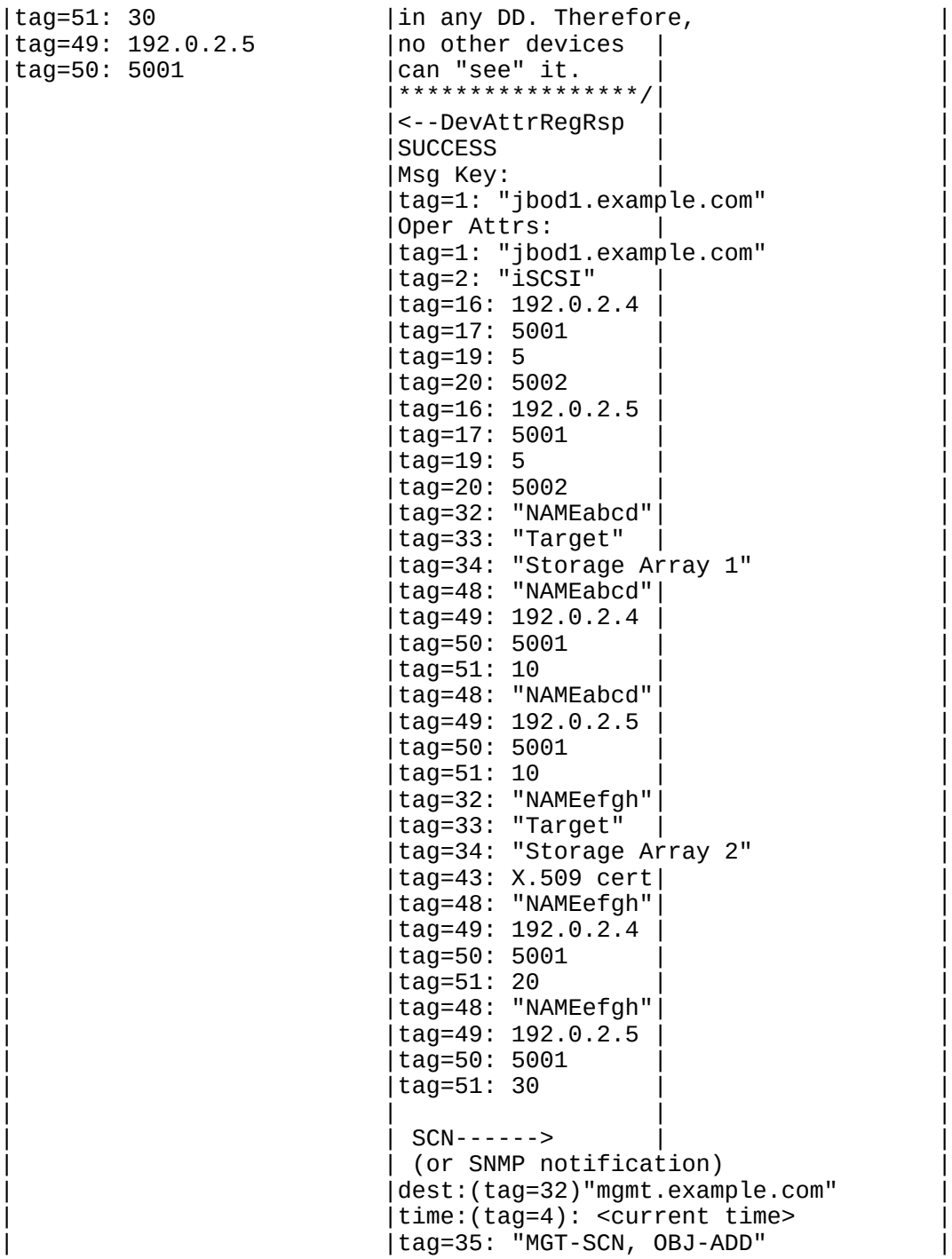


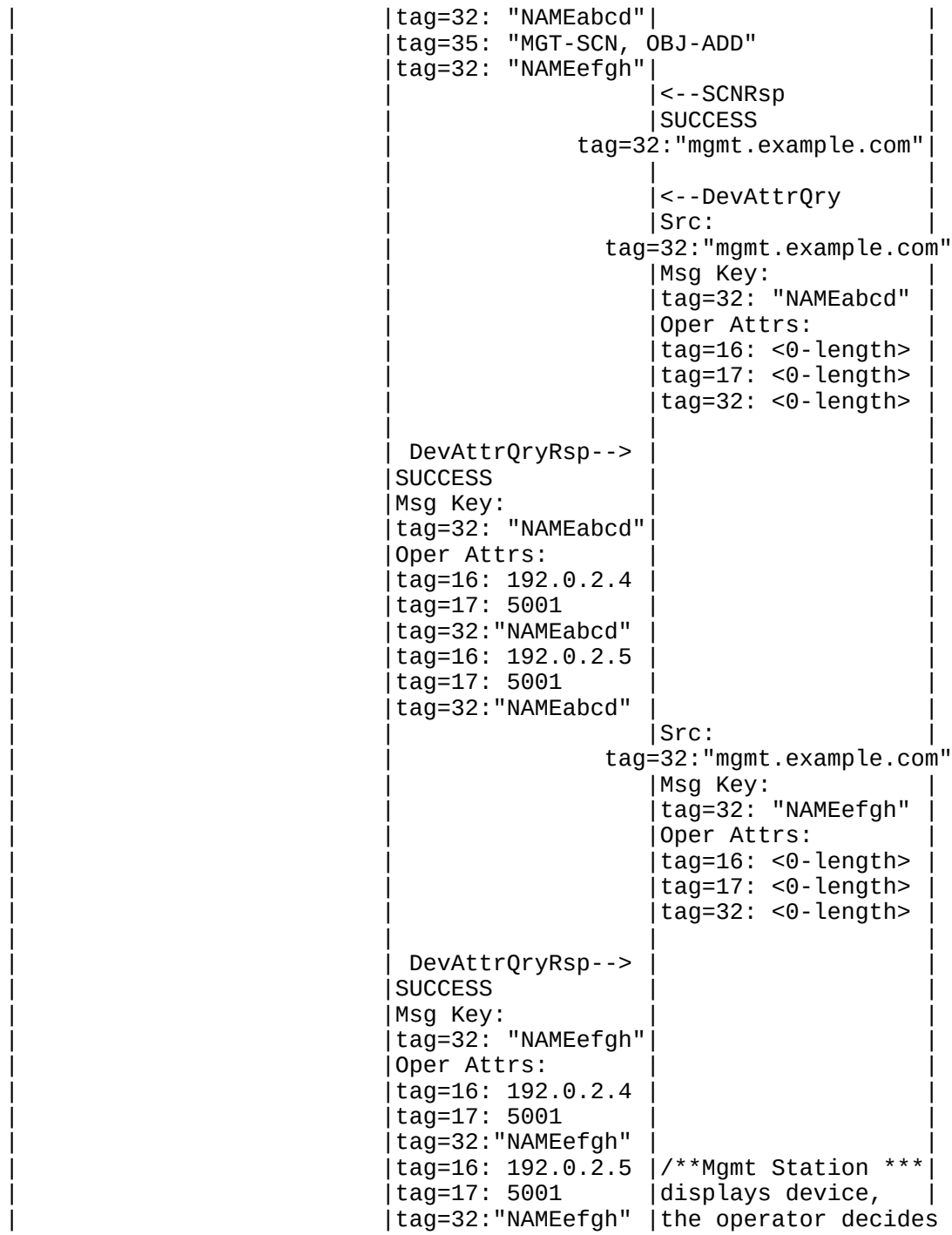

tag=16: 192.0.2.5
tag=17: 5001
tag=32: "NAMEabcd"
tag=33: target
tag=34: "disk 1"
/* previously placed in a DD */
SCN-------->
(or SNMP notification)
dest:(tag=32):"MGMTname1"
time:(tag=4): <current time>
tag=35: "MGT-SCN, OBJ-ADD"
tag=32: "NAMEabcd"
<-------SCNRsp
DevAttrQry--------->
Src:(tag=32) "NAMEabcd"
Key:(tag=33) "initiator"
Oper Attrs:
tag=16: NULL
tag=17: NULL
tag=32: NULL
/*Query asks for all initr
devices' IP address, port
number, and Name*/
<---DevAttrQryRsp
SUCCESS
tag=16:192.0.2.1
tag=17:50000
tag=32:"devpdq"
tag=16:192.0.2.2
tag=17:50000
tag=32:"devrst"
/*************************
Our target "NAMEabcd"
discovers two initiators
in shared DDs. It will
accept iSCSI logins from
these two identified
initiators presented by
iSNS
************************/
<-----DevAttrQry
src: "MGMTname1"
key:(tag=32)"NAMEabcd"
Op Attrs:
tag=16: NULL
tag=17: NULL
tag=32: NULL
DevAttrQryRsp--->
SUCCESS
tag=16: 192.0.2.5
tag=17: 5001
tag=32: "NAMEabcd"

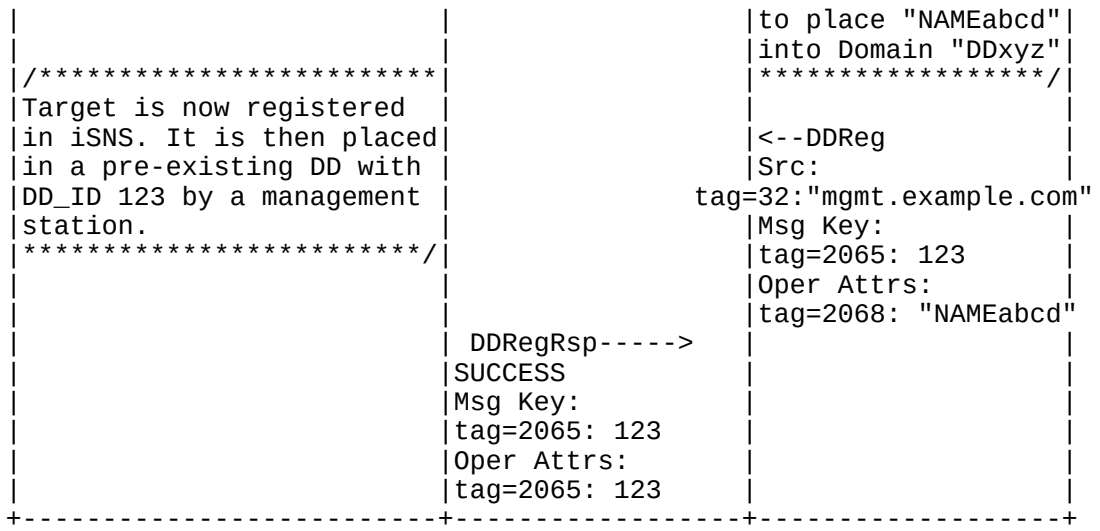
A.1.2. Target Registration and DD Configuration

In this example, a more complex target, with two Storage Nodes and two Portals using ESI monitoring, registers with the iSNS. This target has been configured with a Fully Qualified Domain Name (FQDN) in the DNS servers, and the user wishes to use this identifier for the device. The target explicitly registers Portal Groups to describe how each Portal provides access to each Storage Node. One target Storage Node allows coordinated access through both Portals. The other Storage Node allows access, but not coordinated access, through both Portals.

```
+--------------------------+------------------+--------------------+
|    iSCSI Target Device   |    iSNS Server   |Management Station  |
+--------------------------+------------------+--------------------+
|Discover iSNS--SLP-->     |                  |/*mgmt station is   |
|                          |<--SLP--iSNS Here:| administratively   |
|                          |      192.0.2.100 | authorized to view|
| DevAttrReg-->            |                  | all DDs */         |
|Src:                      |                  |                    |
|tag=32: "NAMEabcd"        |                  |                    |
|Msg Key:                  |                  |                    |
|tag=1: "jbod1.example.com"|                  |                    |
|Oper Attrs:               |                  |                    |
|tag=1: "jbod1.example.com"|                  |                    |
|tag=2: "iSCSI"            |                  |                    |
|tag=16: 192.0.2.4         |                  |                    |
|tag=17: 5001              |                  |                    |
|tag=19: 5                 |                  |                    |
|tag=20: 5002              |                  |                    |
|tag=16: 192.0.2.5         |                  |                    |
|tag=17: 5001              |                  |                    |
|tag=19: 5                 |                  |                    |
|tag=20: 5002              |                  |                    |
|tag=32: "NAMEabcd"        |                  |                    |
|tag=33: "Target"          |                  |                    |
|tag=34: "Storage Array 1" |                  |                    |
|tag=51: 10                |                  |                    |
|tag=49: 192.0.2.4         |                  |                    |
|tag=50: 5001              |                  |                    |
|tag=49: 192.0.2.5         |                  |                    |
|tag=50: 5001              |                  |                    |
|tag=32: "NAMEefgh"        |                  |                    |
|tag=33: "Target"          |                  |                    |
|tag=34: "Storage Array 2" |/*****************|                    |
|tag=51: 20                |jbod1.example.com is                   |
|tag=49: 192.0.2.4         |now registered in |                    |
|tag=50: 5001              |iSNS, but is not  |                    |
```

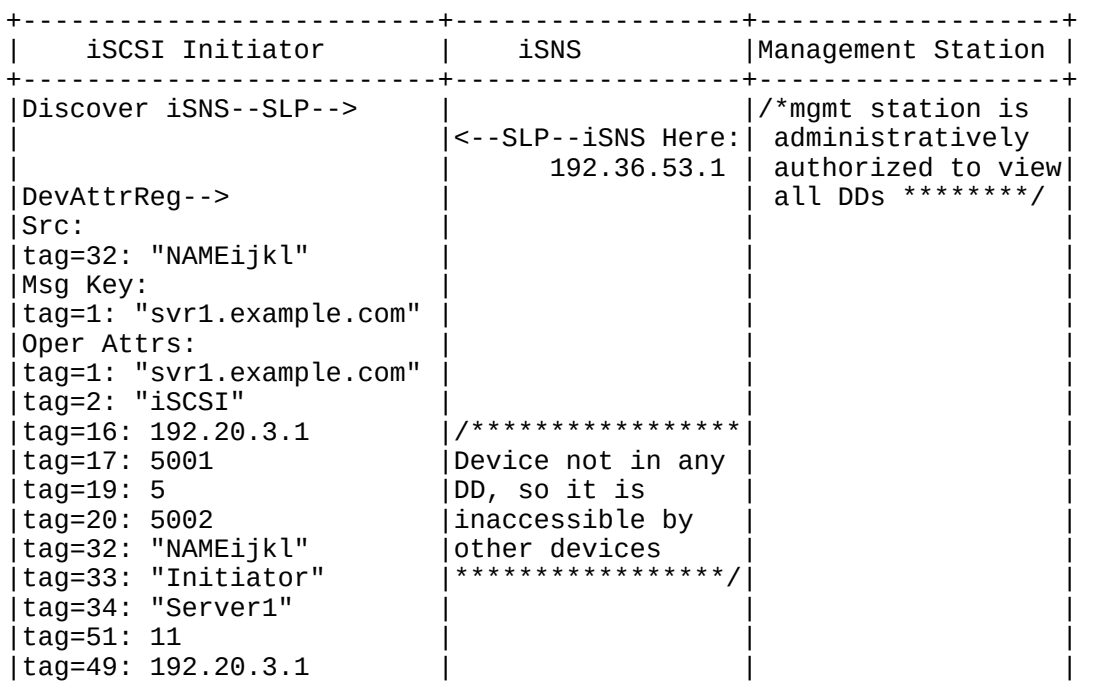

tag=51: 30
tag=49: 192.0.2.5
tag=50: 5001
in any DD. Therefore,
no other devices
can "see" it.
**********/
<--DevAttrRegRsp
SUCCESS
Msg Key:
tag=1: "jbod1.example.com"
Oper Attrs:
tag=1: "jbod1.example.com"
tag=2: "iSCSI"
tag=16: 192.0.2.4
tag=17: 5001
tag=19: 5
tag=20: 5002
tag=16: 192.0.2.5
tag=17: 5001
tag=19: 5
tag=20: 5002
tag=32: "NAMEabcd"
tag=33: "Target"
tag=34: "Storage Array 1"
tag=48: "NAMEabcd"
tag=49: 192.0.2.4
tag=50: 5001
tag=51: 10
tag=48: "NAMEabcd"
tag=49: 192.0.2.5
tag=50: 5001
tag=51: 10
tag=32: "NAMEefgh"
tag=33: "Target"
tag=34: "Storage Array 2"
tag=43: X.509 cert
tag=48: "NAMEefgh"
tag=49: 192.0.2.4
tag=50: 5001
tag=51: 20
tag=48: "NAMEefgh"
tag=49: 192.0.2.5
tag=50: 5001
tag=51: 30
SCN------>
(or SNMP notification)
dest:(tag=32)"mgmt.example.com"
time:(tag=4): <current time>
tag=35: "MGT-SCN, OBJ-ADD"

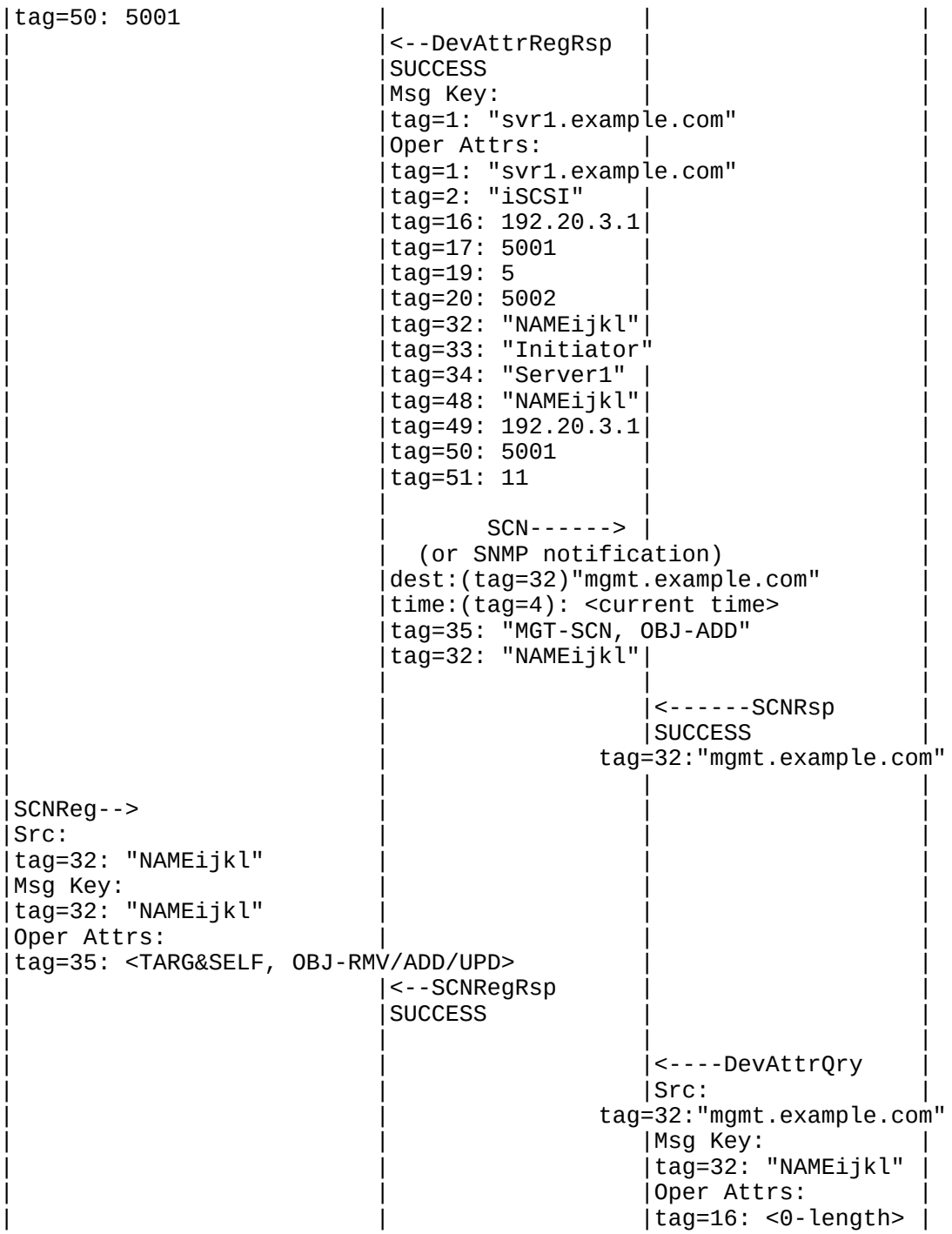

tag=32: "NAMEabcd"
tag=35: "MGT-SCN, OBJ-ADD"
tag=32: "NAMEefgh"
<--SCNRsp
SUCCESS
tag=32:"mgmt.example.com"
<--DevAttrQry
Src:
tag=32:"mgmt.example.com"
Msg Key:
tag=32: "NAMEabcd"
Oper Attrs:
tag=16: <0-length>
tag=17: <0-length>
tag=32: <0-length>
DevAttrQryRsp-->
SUCCESS
Msg Key:
tag=32: "NAMEabcd"
Oper Attrs:
tag=16: 192.0.2.4
tag=17: 5001
tag=32:"NAMEabcd"
tag=16: 192.0.2.5
tag=17: 5001
tag=32:"NAMEabcd"
Src:
tag=32:"mgmt.example.com"
Msg Key:
tag=32: "NAMEefgh"
Oper Attrs:
tag=16: <0-length>
tag=17: <0-length>
tag=32: <0-length>
DevAttrQryRsp-->
SUCCESS
Msg Key:
tag=32: "NAMEefgh"
Oper Attrs:
tag=16: 192.0.2.4
tag=17: 5001
tag=32:"NAMEefgh"
tag=16: 192.0.2.5
tag=17: 5001
tag=32:"NAMEefgh"
/**Mgmt Station ***
displays device,
the operator decides

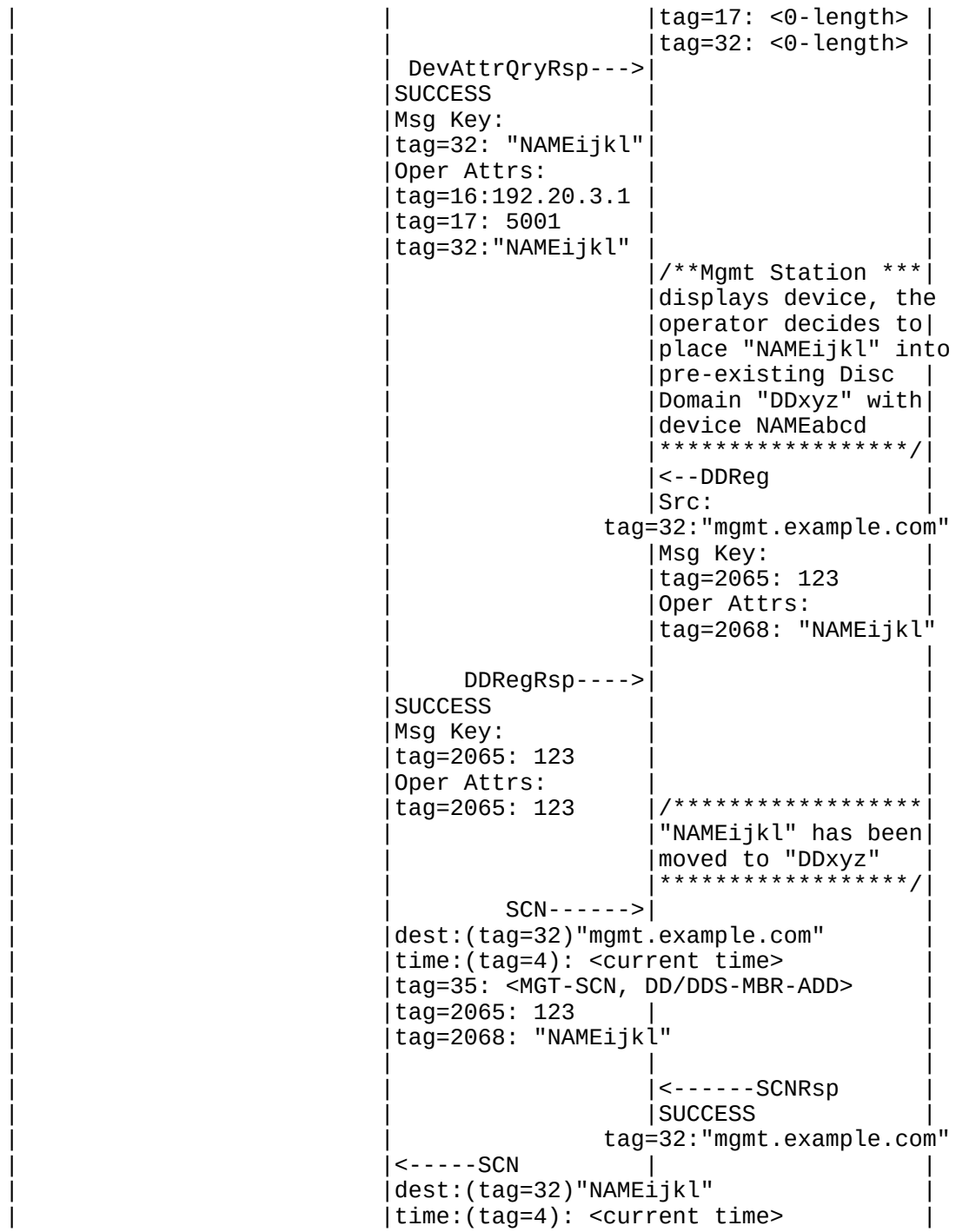


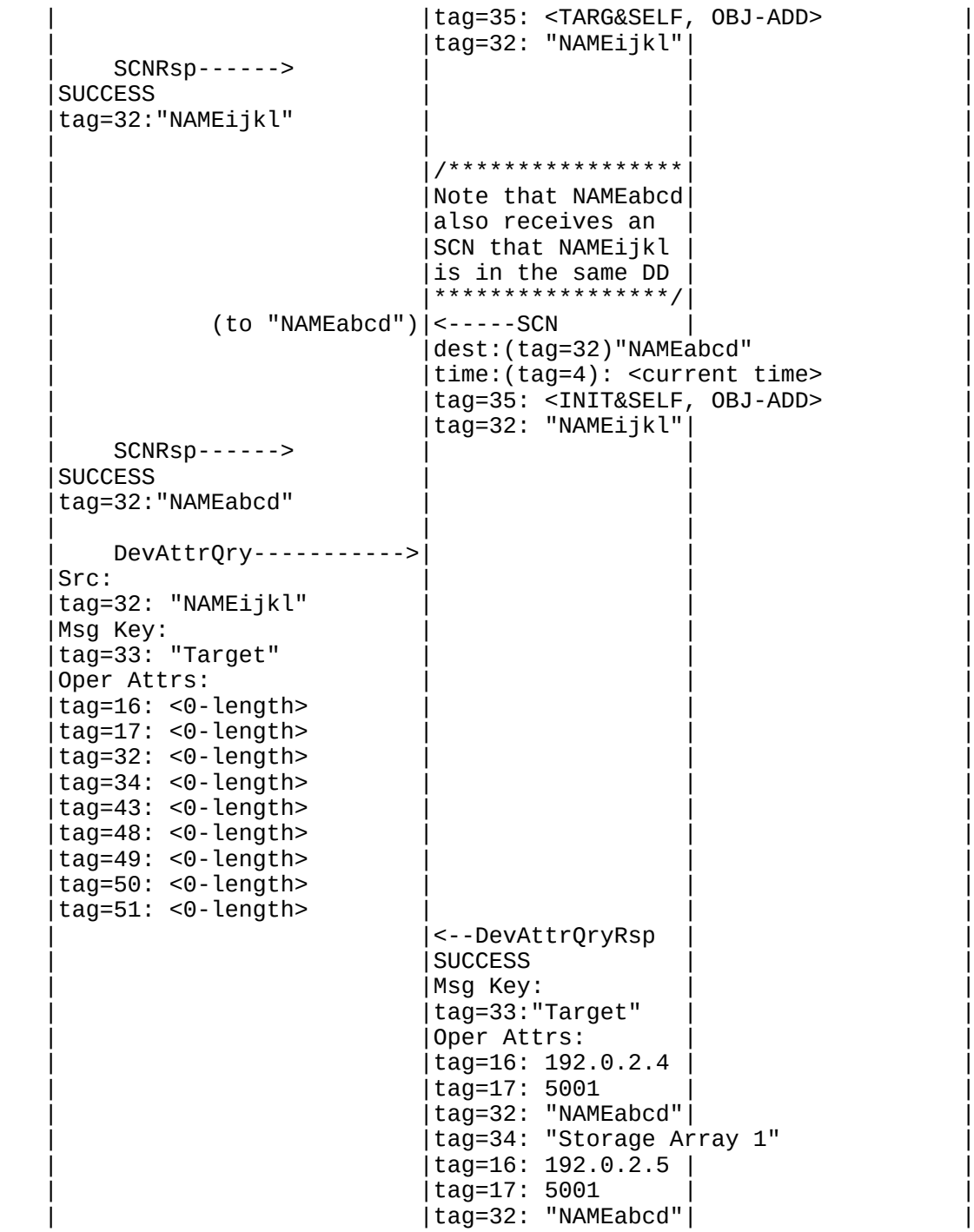
A.1.3. Initiator Registration and Target Discovery

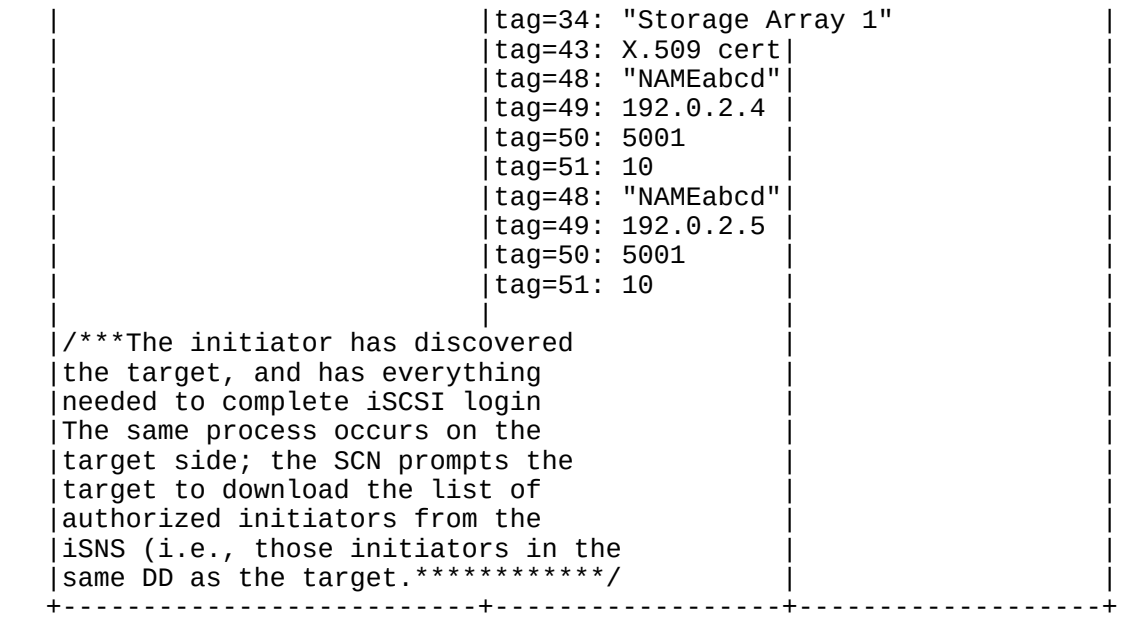
The following example illustrates a new initiator registering with the iSNS, and discovering the target NAMEabcd from the example in A.1.2.




tag=50: 5001
<--DevAttrRegRsp
SUCCESS
Msg Key:
tag=1: "svr1.example.com"
Oper Attrs:
tag=1: "svr1.example.com"
tag=2: "iSCSI"
tag=16: 192.20.3.1
tag=17: 5001
tag=19: 5
tag=20: 5002
tag=32: "NAMEijkl"
tag=33: "Initiator"
tag=34: "Server1"
tag=48: "NAMEijkl"
tag=49: 192.20.3.1
tag=50: 5001
tag=51: 11
SCN------>
(or SNMP notification)
dest:(tag=32)"mgmt.example.com"
time:(tag=4): <current time>
tag=35: "MGT-SCN, OBJ-ADD"
tag=32: "NAMEijkl"
<------SCNRsp
SUCCESS
tag=32:"mgmt.example.com"
SCNReg-->
Src:
tag=32: "NAMEijkl"
Msg Key:
tag=32: "NAMEijkl"
Oper Attrs:
tag=35: <TARG&SELF, OBJ-RMV/ADD/UPD>
<--SCNRegRsp
SUCCESS
<----DevAttrQry
Src:
tag=32:"mgmt.example.com"
Msg Key:
tag=32: "NAMEijkl"
Oper Attrs:
tag=16: <0-length>


tag=17: <0-length>
tag=32: <0-length>
DevAttrQryRsp--->
SUCCESS
Msg Key:
tag=32: "NAMEijkl"
Oper Attrs:
tag=16:192.20.3.1
tag=17: 5001
tag=32:"NAMEijkl"
/**Mgmt Station ***
displays device, the
operator decides to
place "NAMEijkl" into
pre-existing Disc
Domain "DDxyz" with
device NAMEabcd
**********/
<--DDReg
Src:
tag=32:"mgmt.example.com"
Msg Key:
tag=2065: 123
Oper Attrs:
tag=2068: "NAMEijkl"
DDRegRsp---->
SUCCESS
Msg Key:
tag=2065: 123
Oper Attrs:
tag=2065: 123
/**********
"NAMEijkl" has been
moved to "DDxyz"
**********/
SCN------>
dest:(tag=32)"mgmt.example.com"
time:(tag=4): <current time>
tag=35: <MGT-SCN, DD/DDS-MBR-ADD>
tag=2065: 123
tag=2068: "NAMEijkl"
<------SCNRsp
SUCCESS
tag=32:"mgmt.example.com"
<-----SCN
dest:(tag=32)"NAMEijkl"
time:(tag=4): <current time>


tag=35: <TARG&SELF, OBJ-ADD>
tag=32: "NAMEijkl"
SCNRsp------>
SUCCESS
tag=32:"NAMEijkl"
/*****************
Note that NAMEabcd
also receives an
SCN that NAMEijkl
is in the same DD
*****************/
(to "NAMEabcd")
<-----SCN
dest:(tag=32)"NAMEabcd"
time:(tag=4): <current time>
tag=35: <INIT&SELF, OBJ-ADD>
tag=32: "NAMEijkl"
SCNRsp------>
SUCCESS
tag=32:"NAMEabcd"
DevAttrQry----------->
Src:
tag=32: "NAMEijkl"
Msg Key:
tag=33: "Target"
Oper Attrs:
tag=16: <0-length>
tag=17: <0-length>
tag=32: <0-length>
tag=34: <0-length>
tag=43: <0-length>
tag=48: <0-length>
tag=49: <0-length>
tag=50: <0-length>
tag=51: <0-length>
<--DevAttrQryRsp
SUCCESS
Msg Key:
tag=33:"Target"
Oper Attrs:
tag=16: 192.0.2.4
tag=17: 5001
tag=32: "NAMEabcd"
tag=34: "Storage Array 1"
tag=16: 192.0.2.5
tag=17: 5001
tag=32: "NAMEabcd"



Acknowledgements

Numerous individuals contributed to the creation of this document through their careful review and submissions of comments and recommendations. We acknowledge the following persons for their technical contributions to this document: Mark Bakke (Cisco), John Hufferd (IBM), Julian Satran (IBM), Kaladhar Voruganti(IBM), Joe Czap (IBM), John Dowdy (IBM), Tom McSweeney (IBM), Jim Hafner (IBM), Chad Gregory (Intel), Yaron Klein (Sanrad), Larry Lamers (Adaptec), Jack Harwood (EMC), David Black (EMC), David Robinson (Sun), Alan Warwick (Microsoft), Bob Snead (Microsoft), Fa Yoeu (Intransa), Joe White (McDATA), Charles Monia (McDATA), Larry Hofer (McDATA), Ken Hirata (Vixel), Howard Hall (Pirus), Malikarjun Chadalapaka (HP), Marjorie Krueger (HP), Siva Vaddepuri (McDATA), and Vinai Singh (American Megatrends).

Authors' Addresses

Josh Tseng
Riverbed Technology
501 2nd Street, Suite 410
San Francisco, CA 94107

Phone: (650)274-2109
EMail: joshtseng@yahoo.com

Kevin Gibbons
McDATA Corporation
4555 Great America Parkway
Santa Clara, CA 95054-1208

Phone: (408) 567-5765
EMail: kevin.gibbons@mcdata.com

Franco Travostino
Nortel
600 Technology Park Drive
Billerica, MA 01821 USA

Phone: (978) 288-7708
EMail: travos@nortel.com

Curt du Laney
Rincon Research Corporation
101 North Wilmot Road, Suite 101
Tucson AZ 85711

Phone: (520) 519-4409
EMail: cdl@rincon.com

Joe Souza
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399

Phone: (425) 706-3135
EMail: joes@exmsft.com

Full Copyright Statement

Copyright (C) The Internet Society (2005).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.