

Network Working Group  
Request for Comments: 4939  
Category: Standards Track

K. Gibbons  
2Wire, Inc.  
G. Ramkumar  
SnapTell, Inc.  
S. Kipp  
Brocade, Inc.  
July 2007

## Definitions of Managed Objects for iSNS (Internet Storage Name Service)

### 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 IETF Trust (2007).

### Abstract

The iSNS (Internet Storage Name Service) protocol provides storage name service functionality on an IP network that is being used for iSCSI (Internet Small Computer System Interface) or iFCP (Internet Fibre Channel Protocol) storage. This document provides a mechanism to monitor multiple iSNS Servers, including information about registered objects in an iSNS Server.

## Table of Contents

1. The Internet-Standard Management Framework .....	3
2. Introduction .....	3
2.1. Requirement Levels .....	3
3. Technical Description .....	4
3.1. iSNS Registered Objects .....	4
3.2. iSNS MIB Structure .....	5
3.3. iSNS Server Info .....	5
3.3.1. Control Node Information .....	6
3.3.2. Discovery Domain Set (DDS) .....	6
3.3.3. Discovery Domain (DD) .....	6
3.3.4. Registered Storage Objects .....	6
3.3.4.1. Registered Entities .....	6
3.3.4.2. Registered Portals .....	6
3.3.4.3. Registered Portal Groups .....	7
3.3.4.4. Registered iSCSI Nodes .....	7
3.3.4.5. Registered FC Ports .....	7
3.3.4.6. Registered FC Nodes .....	7
3.4. Multiple Server Instances .....	7
3.5. iSNS Notifications .....	7
4. MIB References .....	7
5. MIB Module .....	8
6. IANA Considerations .....	75
7. Security Considerations .....	76
8. Normative References .....	77
9. Informative References .....	78
10. Acknowledgements .....	78

## 1. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIV2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

## 2. Introduction

The iSNS protocol, as described in RFC 4171 [RFC4171], can be used by IP-based storage devices for dynamic registration and discovery of other storage devices in the network. It has the capability to group devices into storage Discovery Domains, and Discovery Domains into Discovery Domain Sets. The iSNS MIB is designed to allow Simple Network Management Protocol (SNMP) to be used to monitor iSNS servers supporting iSCSI [RFC3720] and iFCP [RFC4172].

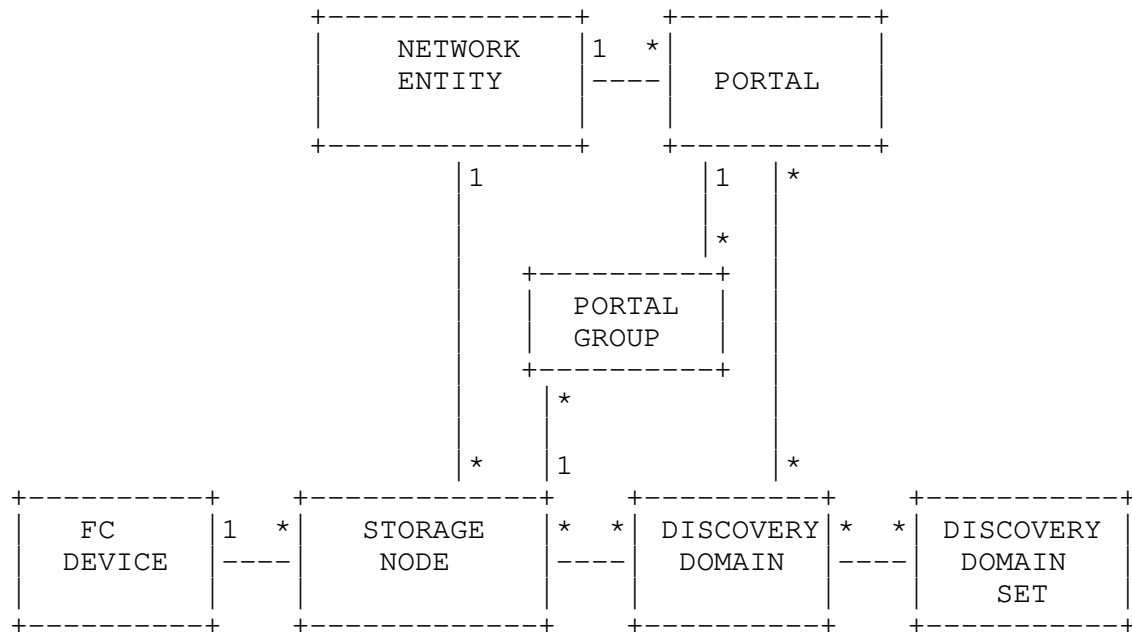
### 2.1. Requirement Levels

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].

### 3. Technical Description

#### 3.1. iSNS Registered Objects

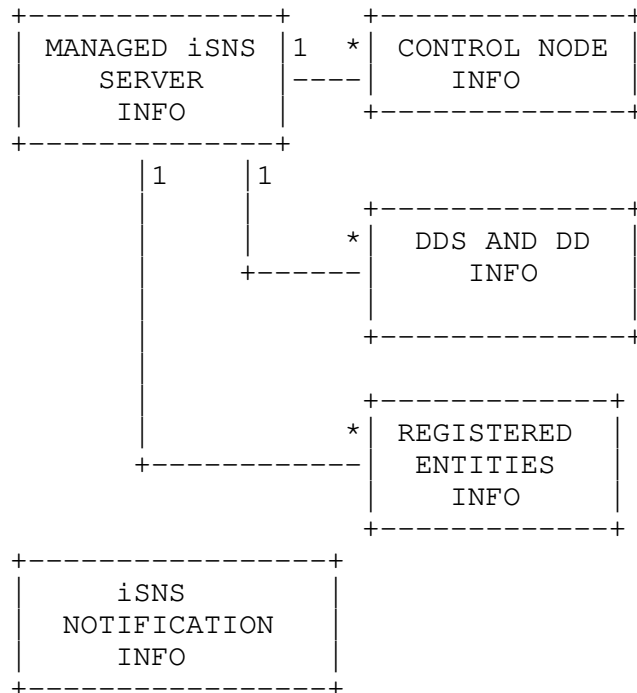
The following entity relationship figure indicates the objects that can be registered in the iSNS, and their relationship to each other.



\* represents 0 to many possible relationships

### 3.2. iSNS MIB Structure

The MIB is divided into sections for iSNS server information, iSNS server registered objects information, and iSNS notifications.



The sections that are required to implement are for iSNS Server management and notification.

### 3.3. iSNS Server Info

The `isnsServerInfo` section provides the ability to monitor multiple iSNS Server instances. The `isnsServerTable` table provides information on each server instance. This table is indexed by the variable `isnsServerIndex`. The table indicates current settings for each iSNS server being managed. The network address, TCP and UDP ports being used by a server for iSNSP registrations and queries can be determined from this table.

The count of objects registered in each iSNS server instance is shown in the table `isnsNumObjectsTable`. The provides a summary of the number Discovery Domain Sets, Discovery Domains, Entities, Portals, Portal Groups, iSCSI Nodes, and iFCP FC Nodes and Ports.

### 3.3.1. Control Node Information

As defined in the iSNS specification, Control Nodes are objects that have been registered with the server and are allowed to manage the iSNS server. These Control Nodes are identified by their iSCSI Node Name or iFCP FC Port Name. The `isnsControlNodeInfo` section of the MIB provides the ability to view the currently registered set of iSCSI and iFCP control nodes.

### 3.3.2. Discovery Domain Set (DDS)

The `isnsDdsInfo` section provides information on each registered DDS, the Discovery Domain members of each DDS, for each iSNS Server instance being managed. DDSs provide a method to group multiple Discovery Domains for easier control. As described in the iSNS Specification [RFC4171], a DDS can be enabled or disabled, which in turn enables or disables the member Discovery Domains. Discovery Domains that are contained in an enabled DDS are then enforced by an iSNS Server.

### 3.3.3. Discovery Domain (DD)

The `isnsDdInfo` section provides information on each registered DD, and the DD members, for each iSNS Server instance being managed. DDs are collections of storage nodes and portals that are allowed to discover one another. DD members can be iSCSI nodes, Entity Portals, or iFCP nodes.

### 3.3.4. Registered Storage Objects

The `isnsReg` section provides information on the registered storage objects for a specific iSNS Server instance. This section is divided into subsections for Entities, Portals, and iSCSI Nodes, as well as iFCP Port and Node information.

#### 3.3.4.1. Registered Entities

The `isnsRegEntityInfo` section provides information on the registered entities. Entities are collections of storage nodes and portals.

#### 3.3.4.2. Registered Portals

The `isnsRegPortalInfo` section provides information on the registered portals for a specific iSNS Server instance. Portals are logical IP-Address, TCP/UDP Port pairs that provide access to storage nodes contained in the associated Entity.

#### 3.3.4.3. Registered Portal Groups

The `isnsRegPortalGroupInfo` section provides information on the registered portal groups for a specific iSNS Server instance. As described in iSCSI [RFC3720], Portal Groups provide a mapping between Portals and iSCSI Storage Nodes contained in an Entity.

#### 3.3.4.4. Registered iSCSI Nodes

The `isnsRegIscsiNodeInfo` section provides information on the registered iSCSI Nodes for a specific iSNS Server instance. The iSCSI nodes are individual storage targets or initiators.

#### 3.3.4.5. Registered FC Ports

The `isnsRegFcPortInfo` section provides information on the registered FC Ports for a specific iSNS Server instance. The FC Ports are ports associated with an iFCP gateway.

#### 3.3.4.6. Registered FC Nodes

The `isnsRegFcNodeInfo` section provides information on the registered FC Nodes for a specific iSNS Server instance. The FC nodes are individual storage devices associated with an iFCP gateway.

### 3.4. Multiple Server Instances

The management of multiple instances of iSNS servers by the agent is supported. As described in Section 3.3, each managed iSNS server instance has an entry in the table `isnsServerTable`.

### 3.5. iSNS Notifications

The `isnsNotification` section provides SNMP notifications for iSNS Server state changes.

## 4. MIB References

The following MIB module has IMPORTS from [RFC2578], [RFC2579], [RFC2580], [RFC3411], [RFC4001], [RFC4044], and [RFC4133]. In REFERENCE clauses, it also refers to [RFC3720], [RFC4171], and [RFC4172].

## 5. MIB Module

```
ISNS-MIB DEFINITIONS ::= BEGIN
  IMPORTS
    -- From RFC 2578
    MODULE-IDENTITY,
    OBJECT-TYPE,
    NOTIFICATION-TYPE,
    Integer32,
    Unsigned32,
    Gauge32,
    mib-2
      FROM SNMPv2-SMI

    -- From RFC 2579
    TEXTUAL-CONVENTION,
    TimeStamp,
    TruthValue
      FROM SNMPv2-TC

    -- From RFC 2580
    OBJECT-GROUP,
    MODULE-COMPLIANCE,
    NOTIFICATION-GROUP
      FROM SNMPv2-CONF

    -- From RFC 3411
    SnmpAdminString
      FROM SNMP-FRAMEWORK-MIB

    -- From RFC 4001
    InetAddressType,
    InetAddress,
    InetPortNumber
      FROM INET-ADDRESS-MIB

    -- From RFC 4044
    FcNameIdOrZero,
    FcAddressIdOrZero
      FROM FC-MGMT-MIB

    -- From RFC 4133
    PhysicalIndex
      FROM ENTITY-MIB
    ;

  isnsMIB MODULE-IDENTITY
    LAST-UPDATED "200707110000Z"
```



ORGANIZATION "IETF IPS Working Group"  
CONTACT-INFO "

Attn: Kevin Gibbons  
2Wire, Inc.  
1704 Automation Parkway  
San Jose, CA 95131  
USA  
Tel: +1 408-895-1387  
Fax: +1 408-428-9590  
Email: kgibbons@yahoo.com

G.D. Ramkumar  
SnapTell, Inc.  
2741 Middlefield Rd, Suite 200  
Palo Alto, CA 94306  
USA  
Tel: +1 650-326-7627  
Fax: +1 650-326-7620  
Email: gramkumar@stanfordalumni.org

Scott Kipp  
Brocade  
4 McDATA Pkwy  
Broomfield, CO 80021  
USA  
Tel: +1 720-558-3452  
Fax: +1 720-558-8999  
Email: skip@brocade.com

"

DESCRIPTION

"This module defines management information  
specific to internet Storage Name Service  
(iSNS) management.

Copyright (C) The IETF Trust (2007).  
This version of this MIB module is part  
of RFC 4939; see the RFC itself for full  
legal notices."

REVISION "200707110000Z"

DESCRIPTION

"Initial version of iSNS Management Module.  
This MIB published as RFC 4939."

::= { mib-2 163 }

```
--
--  Textual Conventions
--
```

IsnsDiscoveryDomainSetId ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUS current

DESCRIPTION

"The unique Discovery Domain Set Identifier associated with a Discovery Domain Set (DDS)."

REFERENCE "RFC 4171, Section 6.11.1.1"

SYNTAX Unsigned32 ( 1 .. 4294967295 )

IsnsDdsStatusType ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"The status of a Discovery Domain Set (DDS) registered in the iSNS. The initially assigned values are below:

Bit	Status
-----	-----
31	DDS Enabled
All others	RESERVED

Setting a bit to 1 indicates the feature is enabled. Otherwise, it is disabled. The future assignment of any of the reserved values will be documented in a revision of RFC 4171."

REFERENCE "RFC 4171, Section 6.11.1.3"

SYNTAX BITS {

reserved0(0), reserved1(1), reserved2(2),

reserved3(3), reserved4(4), reserved5(5),

reserved6(6), reserved7(7), reserved8(8),

reserved9(9), reserved10(10), reserved11(11),

reserved12(12), reserved13(13), reserved14(14),

reserved15(15), reserved16(16), reserved17(17),

reserved18(18), reserved19(19), reserved20(20),

reserved21(21), reserved22(22), reserved23(23),

reserved24(24), reserved25(25), reserved26(26),

reserved27(27), reserved28(28), reserved29(29),

reserved30(30),

ddsEnabled (31)

}

IsnsDiscoveryDomainId ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUS current

DESCRIPTION

"The unique Discovery Domain Identifier (DD\_ID) associated

with each Discovery Domain (DD). This is used to uniquely index and reference a DD."

REFERENCE "RFC 4171, Section 6"

SYNTAX Unsigned32 ( 1 .. 4294967295 )

IsnsDdFeatureType ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"This type defines the features that each Discovery Domain (DD) has.

Bit	Status
-----	-----
31	Boot List
All others	RESERVED

Boot List: this feature indicates that the targets in this DD provide boot capabilities for the member initiators.

Setting a bit to 1 indicates the feature is enabled. Otherwise, it is disabled. The future assignment of any of the reserved values will be documented in a revision of RFC 4171."

REFERENCE "RFC 4171, Section 6.11.2.9"

SYNTAX BITS {  
 reserved0(0), reserved1(1), reserved2(2),  
 reserved3(3), reserved4(4), reserved5(5),  
 reserved6(6), reserved7(7), reserved8(8),  
 reserved9(9), reserved10(10), reserved11(11),  
 reserved12(12), reserved13(13), reserved14(14),  
 reserved15(15), reserved16(16), reserved17(17),  
 reserved18(18), reserved19(19), reserved20(20),  
 reserved21(21), reserved22(22), reserved23(23),  
 reserved24(24), reserved25(25), reserved26(26),  
 reserved27(27), reserved28(28), reserved29(29),  
 reserved30(30),  
 bootlist(31)  
 }

IsnsDdDdsModificationType ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"The methods that can be used to modify the Discovery Domain and Discovery Domain Sets in an iSNS Server instance.

Bit	Flag Description
-----	-----
0	Control Nodes are allowed

- 1 Target iSCSI Nodes are allowed
- 2 Initiator iSCSI Nodes are allowed
- 3 Target iFCP Ports are allowed
- 4 Initiator iFCP Ports are allowed

Setting a bit to 1 indicates the feature is enabled. Otherwise, it is disabled."

REFERENCE "RFC 4171, Section 2.4"

SYNTAX BITS {  
     controlNode(0),  
     targetIscsiNode(1),  
     initiatorIscsiNode(2),  
     targetIfcpNode(3),  
     initiatorIfcpNode(4)  
   }

IsnsEntityIndexIdOrZero ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUS current

DESCRIPTION

"The identifier for the unique integer Entity Index associated with an iSNS registered Entity object, and the value zero. The value zero is object-specific and MUST therefore be defined as part of the description of any object that uses this syntax. Examples of the usage of zero might include situations where the Entity is unknown, or not yet registered in the iSNS server. If a value of zero is not valid for an object, then that MUST be indicated."

REFERENCE "RFC 4171, Section 6"

SYNTAX Unsigned32 ( 0 .. 4294967295 )

IsnsPortalGroupIndexId ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUS current

DESCRIPTION

"The identifier for the unique integer Portal Group Index associated with an iSNS registered Portal Group object."

REFERENCE "RFC 4171, Section 6"

SYNTAX Unsigned32 ( 1 .. 4294967295 )

IsnsPortalIndexId ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUS current

DESCRIPTION

"The identifier for the unique integer Portal Index associated with an iSNS registered Portal object. The index is created by the iSNS Server for mapping between

registered objects. The Portal Index used for a specific portal IP-address and port number pair is only persistent across reboots for portals that have been explicitly added to a Discovery Domain (DD). If a portal is not explicitly registered in any DD, then the index used for a portal can change after a server reinitialization."

REFERENCE "RFC 4171, Section 6"

SYNTAX Unsigned32 ( 1 .. 4294967295 )

IsnsPortalPortTypeId ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"The UDP or TCP port type being used by a Portal for an Entity."

REFERENCE "RFC 4171, Section 6.3.2"

SYNTAX INTEGER { udp(1), tcp(2) }

IsnsPortalGroupTagIdOrNull ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUS current

DESCRIPTION

"The Portal Group Tag (PGT) represents an association between a Portal and iSCSI Node using the value range 0 to 65535. A PGT with no association is a NULL value. The value of -1 indicates a NULL value."

REFERENCE "RFC 4171, Section 6.5.4, and RFC 3720"

SYNTAX Integer32 ( -1 .. 65535 )

IsnsPortalSecurityType ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"Indicates security attribute settings for a Portal that is registered in the iSNS server. The bitmapVALID field must be set in order for the contents to be considered valid information. The definitions of the bit fields are based on RFC 4171. The initial representation of each bit setting (0 or 1) is indicated below.

Bit	Flag Description
-----	-----
25	1 = Tunnel Mode Preferred; 0 = No Preference
26	1 = Transport Mode Preferred; 0 = No Preference
27	1 = 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	1 = Bitmap VALID; 0 = INVALID

All others RESERVED

The future assignment of any of the reserved values will be documented in a revision of RFC 4171."

REFERENCE "RFC 4171, Section 6.3.9"

SYNTAX BITS {  
     reserved0(0), reserved1(1), reserved2(2),  
     reserved3(3), reserved4(4), reserved5(5),  
     reserved6(6), reserved7(7), reserved8(8),  
     reserved9(9), reserved10(10), reserved11(11),  
     reserved12(12), reserved13(13), reserved14(14),  
     reserved15(15), reserved16(16), reserved17(17),  
     reserved18(18), reserved19(19), reserved20(20),  
     reserved21(21), reserved22(22), reserved23(23),  
     reserved24(24),  
     tunnelModePreferred(25),  
     transportModePreferred(26),  
     pfsEnabled(27),  
     agressiveModeEnabled(28),  
     mainModeEnabled(29),  
     ikeIPsecEnabled(30),  
     bitmapVALID(31)  
 }

IsnsNodeIndexId ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUS current

DESCRIPTION

"The identifier for the unique integer Node Index associated with a storage node. This index provides a 1-to-1 mapping to an iSCSI node name. The iSCSI node name maximum length is too long to be used for an index directly. The iSCSI node index used for a specific iSCSI node name is identical in all DDs, and is persistent across server reinitializations when the iSCSI node is a member of a Discovery Domain (DD) or is registered as a Control Node. Furthermore, index values for recently deregistered objects SHOULD NOT be reused in the short term."

REFERENCE "RFC 4171, Section 6.4.5"

SYNTAX Unsigned32 ( 1 .. 4294967295 )

IsnsIscsiNodeType ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"The iSCSI Node Type defines the functions of the registered object. The definitions of each setting are defined in RFC 4171.

Bit	Node Type
-----	-----------

-----	-----
29	Control
30	Initiator
31	Target
All others	RESERVED

Setting a bit to 1 indicates the node has the corresponding characteristics. The future assignment of any of the reserved values will be documented in a revision of RFC 4171."

REFERENCE "RFC 4171, Section 6.4.2"

SYNTAX BITS {  
 reserved0(0), reserved1(1), reserved2(2),  
 reserved3(3), reserved4(4), reserved5(5),  
 reserved6(6), reserved7(7), reserved8(8),  
 reserved9(9), reserved10(10), reserved11(11),  
 reserved12(12), reserved13(13), reserved14(14),  
 reserved15(15), reserved16(16), reserved17(17),  
 reserved18(18), reserved19(19), reserved20(20),  
 reserved21(21), reserved22(22), reserved23(23),  
 reserved24(24), reserved25(25), reserved26(26),  
 reserved27(27), reserved28(28),  
 control(29),  
 initiator(30),  
 target(31)  
 }

IsnsFcClassOfServiceType ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"This defines the Fibre Channel Class of Service types that are supported by the registered port. The definitions are as defined in RFC 4171.

Bit	FC COS Type
-----	-----
28	Fibre Channel Class 3 Supported
29	Fibre Channel Class 2 Supported
All others	RESERVED

Setting a bit to 1 indicates the class of service is supported. The future assignment of any of the reserved values will be documented in a revision of RFC 4171."

REFERENCE "RFC 4171, Section 6.6.8"

SYNTAX BITS {  
 reserved0(0), reserved1(1), reserved2(2),  
 reserved3(3), reserved4(4), reserved5(5),  
 reserved6(6), reserved7(7), reserved8(8),

```

reserved9(9), reserved10(10), reserved11(11),
reserved12(12), reserved13(13), reserved14(14),
reserved15(15), reserved16(16), reserved17(17),
reserved18(18), reserved19(19), reserved20(20),
reserved21(21), reserved22(22), reserved23(23),
reserved24(24), reserved25(25), reserved26(26),
reserved27(27),
class3(28),
class2(29)
}

```

IsnsIscsiScnType ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"The iSCSI Node State Change Notification (SCN) values for a node as defined in RFC 4171.

Bit	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 or DDS member removed (Mgmt Reg/SCN only)
31 (Lsb)	DD or DDS member added (Mgmt Reg/SCN only)
All others	Reserved

Setting a bit to 1 indicates that type of SCN is enabled. The future assignment of any of the reserved values will be documented in a revision of RFC 4171."

REFERENCE "RFC 4171, Section 6.4.4"

SYNTAX BITS {

```

reserved0(0), reserved1(1), reserved2(2),
reserved3(3), reserved4(4), reserved5(5),
reserved6(6), reserved7(7), reserved8(8),
reserved9(9), reserved10(10), reserved11(11),
reserved12(12), reserved13(13), reserved14(14),
reserved15(15), reserved16(16), reserved17(17),
reserved18(18), reserved19(19), reserved20(20),
reserved21(21), reserved22(22), reserved23(23),
initiatorAndSelfOnly(24),
targetAndSelfOnly(25),
managementRegistrationScn(26),
objectRemoved(27),
objectAdded(28),

```



```

    objectUpdated(29),
    ddOrDdsMemberRemoved(30),
    ddOrDdsMemberAdded(31)
}

```

IsnsIfcpScnType ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"The iFCP State Change Notification (SCN) values for an iFCP object as defined in RFC 4171.

Bit	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 or DDS member removed (Mgmt Reg/SCN only)
31 (Lsb)	DD or DDS member added (Mgmt Reg/SCN only)
All others	Reserved

Setting a bit to 1 indicates that type of SCN is enabled. The future assignment of any of the reserved values will be documented in a revision of RFC 4171."

REFERENCE "RFC 4171, Section 6.6.12"

```

SYNTAX BITS {
    reserved0(0), reserved1(1), reserved2(2),
    reserved3(3), reserved4(4), reserved5(5),
    reserved6(6), reserved7(7), reserved8(8),
    reserved9(9), reserved10(10), reserved11(11),
    reserved12(12), reserved13(13), reserved14(14),
    reserved15(15), reserved16(16), reserved17(17),
    reserved18(18), reserved19(19), reserved20(20),
    reserved21(21), reserved22(22), reserved23(23),
    initiatorAndSelfOnly(24),
    targetAndSelfOnly(25),
    managementRegistrationScn(26),
    objectRemoved(27),
    objectAdded(28),
    objectUpdated(29),
    ddOrDdsMemberRemoved(30),
    ddOrDdsMemberAdded(31)
}

```

IsnsFcPortRoleType ::= TEXTUAL-CONVENTION

STATUS                   current  
DESCRIPTION  
"The FC Port Role defines the functions of the registered object. The definitions of each setting are defined in RFC 4171."

Bit -----	Port Role -----
29	Control
30	FCP Initiator
31	FCP Target
All others	RESERVED

Setting a bit to 1 indicates the port has the corresponding characteristics. The future assignment of any of the reserved values will be documented in a revision of RFC 4171."

REFERENCE               "RFC 4171, Section 6.6.13"

SYNTAX                 BITS {  
     reserved0(0), reserved1(1), reserved2(2),  
     reserved3(3), reserved4(4), reserved5(5),  
     reserved6(6), reserved7(7), reserved8(8),  
     reserved9(9), reserved10(10), reserved11(11),  
     reserved12(12), reserved13(13), reserved14(14),  
     reserved15(15), reserved16(16), reserved17(17),  
     reserved18(18), reserved19(19), reserved20(20),  
     reserved21(21), reserved22(22), reserved23(23),  
     reserved24(24), reserved25(25), reserved26(26),  
     reserved27(27), reserved28(28),  
     control(29),  
     initiator(30),  
     target(31)  
     }

IsnsSrvrDiscoveryMethodsType ::= TEXTUAL-CONVENTION

STATUS                   current

DESCRIPTION

"The types of iSNS Server discovery methods that are enabled on an iSNS Server. The options are DHCP, Service Location Protocol (SLP), multicast group iSNS heartbeat, broadcast group iSNS heartbeat, configured server list, and other. The iSNS Server may support additional discovery methods not indicated."

REFERENCE               "RFC 4171, Section 2.5"

SYNTAX                 BITS {  
     dhcp(0),  
     slp(1),  
     multicastGroupHb(2),  
     broadcastHb(3),

```

        cfgdServerList(4),
        other(5)
    }

--
-- Internet Storage Name Service Management
--

isnsNotifications          OBJECT IDENTIFIER ::=
    { isnsMIB 0 }
isnsObjects                OBJECT IDENTIFIER ::=
    { isnsMIB 1 }
isnsConformance            OBJECT IDENTIFIER ::=
    { isnsMIB 2 }

--
-- iSNS Server instance managed objects -----
--

isnsServerInfo OBJECT IDENTIFIER ::= { isnsObjects 1 }

isnsServerTable            OBJECT-TYPE
    SYNTAX                  SEQUENCE OF IsnsServerEntry
    MAX-ACCESS              not-accessible
    STATUS                  current
    DESCRIPTION
        "This table provides a list of the iSNS Server instances
        that are managed through the same SNMP context."
    ::= { isnsServerInfo 1 }

isnsServerEntry            OBJECT-TYPE
    SYNTAX                  IsnsServerEntry
    MAX-ACCESS              not-accessible
    STATUS                  current
    DESCRIPTION
        "This is a row in the iSNS Server instance table. The number
        of rows is dependent on the number of iSNS Server instances
        that are being managed through the same SNMP context."
    INDEX { isnsServerIndex }
    ::= { isnsServerTable 1 }

IsnsServerEntry ::=
    SEQUENCE {
        isnsServerIndex      Unsigned32,
        isnsServerName       SnmpAdminString,
        isnsServerIsnsVersion Unsigned32,
        isnsServerVendorInfo SnmpAdminString,

```

```

    isnsServerPhysicalIndex    PhysicalIndex,
    isnsServerTcpPort          InetPortNumber,
    isnsServerUdpPort          InetPortNumber,
    isnsServerDiscontinuityTime
                                TimeStamp,
    isnsServerRole              INTEGER,
    isnsServerDiscoveryMethodsEnabled
                                IsnsSrvrDiscoveryMethodsType,
    isnsServerDiscoveryMcGroupType
                                InetAddressType,
    isnsServerDiscoveryMcGroupAddress
                                InetAddress,
    isnsServerEsiNonResponseThreshold
                                Unsigned32,
    isnsServerEnableControlNodeMgtScn
                                TruthValue,
    isnsServerDefaultDdDdsStatus
                                INTEGER,
    isnsServerUpdateDdDdsSupported
                                IsnsDdDdsModificationType,
    isnsServerUpdateDdDdsEnabled
                                IsnsDdDdsModificationType
    }

isnsServerIndex                OBJECT-TYPE
    SYNTAX                      Unsigned32 ( 1 .. 4294967295 )
    MAX-ACCESS                  not-accessible
    STATUS                      current
    DESCRIPTION
        "This object uniquely identifies the iSNS Server being
        managed by the SNMP context and is the key for this table.
        This is an instance index for each iSNS Server being
        managed. The value of this object is used elsewhere in
        the MIB to reference specific iSNS Servers."
    ::= { isnsServerEntry 1 }

isnsServerName                 OBJECT-TYPE
    SYNTAX                      SnmpAdminString
    MAX-ACCESS                  read-only
    STATUS                      current
    DESCRIPTION
        "A non-unique name that can be assigned to the iSNS Server
        instance. If not configured, then the string SHALL be
        zero-length."
    ::= { isnsServerEntry 2 }

isnsServerIsnsVersion          OBJECT-TYPE
    SYNTAX                      Unsigned32 ( 0 .. 65535 )

```

MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION

"The iSNS version value as contained in messages received from the current primary server. The header of each iSNSP message contains the iSNS version of the sender. If unknown, the reported value is 0."

REFERENCE "RFC 4171"  
 DEFVAL { 1 }  
 ::= { isnsServerEntry 3 }

isnsServerVendorInfo OBJECT-TYPE  
 SYNTAX SnmpAdminString  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION

"If this server instance is utilizing the product of a particular 'vendor', then this managed object contains that vendor's name and version. Otherwise, the string SHALL be zero-length. The format of the string is as follows: Vendor Name, Vendor Version, Vendor Defined Information.

Field	Description
-----	-----
Vendor Name	The name of the vendor (if one exists)
Vendor Version	The version of the vendor product
Vendor Defined	This follows the second comma in the string, if one exists, and is vendor defined

"  
 ::= { isnsServerEntry 4 }

isnsServerPhysicalIndex OBJECT-TYPE  
 SYNTAX PhysicalIndex  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION

"An index identifying the network interface for this iSNS Server within a network entity. This index maps to the entPhysicalIndex of entPhysicalTable table in RFC 4133. The entPhysicalClass value for the table row must be 'port', as the interface must be able to send and receive data."

REFERENCE "RFC 4133, RFC 4171, Section 2.5 - 2.8"  
 ::= { isnsServerEntry 5 }

isnsServerTcpPort OBJECT-TYPE  
 SYNTAX InetPortNumber

```

    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
    "Indicates the TCP port this iSNS instance is accepting
    iSNSP messages on, generally the iSNS well-known port.
    The well-known TCP port for iSNSP is 3205. If TCP is
    not supported by this server instance, then the value
    is 0."
    ::= { isnsServerEntry 6 }

isnsServerUdpPort      OBJECT-TYPE
    SYNTAX            InetPortNumber
    MAX-ACCESS        read-only
    STATUS            current
    DESCRIPTION
    "Indicates the UDP port this iSNS instance is accepting
    iSNSP messages on; generally, the iSNS well-known port.
    The well-known UDP port for iSNSP is 3205. If UDP is
    not supported by this server instance, then the value
    is 0."
    ::= { isnsServerEntry 7 }

isnsServerDiscontinuityTime OBJECT-TYPE
    SYNTAX            TimeStamp
    MAX-ACCESS        read-only
    STATUS            current
    DESCRIPTION
    "The value of sysUpTime on the most recent occasion that
    this iSNS server became active or suffered a
    discontinuity."
    ::= { isnsServerEntry 8 }

isnsServerRole          OBJECT-TYPE
    SYNTAX              INTEGER { notSet(1),
                                server(2),
                                backupServer(3) }
    MAX-ACCESS          read-only
    STATUS              current
    DESCRIPTION
    "The current operational mode of this iSNS Server instance.

    Value              Description
    -----
    notSet              The iSNS Server role is not
                        configured.
    server              The iSNS Server instance is
                        an operational iSNS Server.
    backupServer        The iSNS Server instance is

```

currently acting as a backup."

REFERENCE "RFC 4171, Section 2.7 - 2.8"

::= { isnsServerEntry 9 }

isnsServerDiscoveryMethodsEnabled OBJECT-TYPE

SYNTAX IsnsSrvrDiscoveryMethodsType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates the discovery methods currently enabled for this iSNS Server instance. This allows a client to determine what discovery methods can be used for this iSNS Server. Additional methods of discovery may also be supported."

::= { isnsServerEntry 10 }

isnsServerDiscoveryMcGroupType OBJECT-TYPE

SYNTAX InetAddressType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The type of Internet address in isnsServerDiscoveryMcGroupAddress. If the address is specified, then it must be a valid multicast address and the value of this object must be ipv4(1), ipv6(2), ipv4z(3), or ipv6z(4); otherwise, the value of this object is unknown(0), and the value of isnsServerDiscoveryMcGroupAddress is the zero-length string."

::= { isnsServerEntry 11 }

isnsServerDiscoveryMcGroupAddress OBJECT-TYPE

SYNTAX InetAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The multicast group that iSNS Heartbeat messages are sent to if multicast-based discovery has been enabled for this server instance. If not configured, then the string SHALL be zero-length. The format of this object is specified by isnsServerDiscoveryMcGroupType."

::= { isnsServerEntry 12 }

isnsServerEsiNonResponseThreshold OBJECT-TYPE

SYNTAX Unsigned32 ( 0 .. 65535 )

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Entity Status Inquiry (ESI) Non-Response Threshold -

the number of ESI messages that will be sent without receiving a response before an entity is deregistered from the iSNS database. A value of 0 indicates Entities will never be deregistered due to non-receipt of ESI messages."

REFERENCE "RFC 4171, Section 2.4"

DEFVAL { 3 }

::= { isnsServerEntry 13 }

isnsServerEnableControlNodeMgtScn OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates if the iSNS Server administrative option to send Management SCNs to Control Nodes is enabled. Management SCNs are used by Control Nodes to monitor and control an iSNS Server. If enabled, Control Nodes can register to receive Management SCNs."

REFERENCE "RFC 4171, Section 2.2.3, 2.4"

DEFVAL { true }

::= { isnsServerEntry 14 }

isnsServerDefaultDdDdsStatus OBJECT-TYPE

SYNTAX INTEGER { inNoDomain(1),  
inDefaultDdAndDds(2) }

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This indicates the Discovery Domain (DD) and Discovery Domain Set (DDS) membership status for a new device when registered in the iSNS Server instance. Either the new device will not be in a DD/DDS, or will be placed into a default DD and default DDS. The default setting is inNoDomain."

REFERENCE "RFC 4171, Section 2.4"

DEFVAL { inNoDomain }

::= { isnsServerEntry 15 }

isnsServerUpdateDdDdsSupported OBJECT-TYPE

SYNTAX IsnsDdDdsModificationType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The methods that this iSNS Server instance supports to modify Discovery Domains and Discovery Domain Sets."

REFERENCE "RFC 4171, Section 2.4"

::= { isnsServerEntry 16 }



```

isnsServerUpdateDdDdsEnabled OBJECT-TYPE
    SYNTAX                IsnsDdDdsModificationType
    MAX-ACCESS             read-only
    STATUS                 current
    DESCRIPTION
        "This indicates the methods this server instance currently
        allows for modifying Discovery Domains and Discovery
        Domain Sets."
    REFERENCE              "RFC 4171, Sec 2.2.2 and 2.4"
    ::= { isnsServerEntry 17 }

--
-- Count of objects currently registered in a server instance
--

isnsNumObjectsTable      OBJECT-TYPE
    SYNTAX                SEQUENCE OF
                        IsnsNumObjectsEntry
    MAX-ACCESS             not-accessible
    STATUS                 current
    DESCRIPTION
        "Table providing the number of registered objects of each
        type in the iSNS Server instance. The number of entries is
        dependent upon the number of iSNS Server instances being
        managed."
    ::= { isnsServerInfo 2 }

isnsNumObjectsEntry      OBJECT-TYPE
    SYNTAX                IsnsNumObjectsEntry
    MAX-ACCESS             not-accessible
    STATUS                 current
    DESCRIPTION
        "Entry of an iSNS Server instance."
    AUGMENTS { isnsServerEntry }
    ::= { isnsNumObjectsTable 1 }

IsnsNumObjectsEntry ::= SEQUENCE {
    isnsNumDds             Gauge32,
    isnsNumDd              Gauge32,
    isnsNumEntities        Gauge32,
    isnsNumPortals         Gauge32,
    isnsNumPortalGroups    Gauge32,
    isnsNumIscsiNodes      Gauge32,
    isnsNumFcPorts         Gauge32,
    isnsNumFcNodes         Gauge32
}

```

```

isnsNumDds                OBJECT-TYPE
    SYNTAX                 Gauge32 ( 0 .. 4294967295 )
    MAX-ACCESS              read-only
    STATUS                  current
    DESCRIPTION
        "The current total number of Discovery Domain Sets
        in this iSNS instance.  This is the number of rows
        in the isnsDdsTable."
        ::= { isnsNumObjectsEntry 1 }

isnsNumDd                  OBJECT-TYPE
    SYNTAX                 Gauge32 ( 0 .. 4294967295 )
    MAX-ACCESS              read-only
    STATUS                  current
    DESCRIPTION
        "The current total number of Discovery Domains
        in this iSNS instance.  This is the number of rows in the
        isnsDdTable."
        ::= { isnsNumObjectsEntry 2 }

isnsNumEntities            OBJECT-TYPE
    SYNTAX                 Gauge32 ( 0 .. 4294967295 )
    MAX-ACCESS              read-only
    STATUS                  current
    DESCRIPTION
        "The current number of Entities registered in this
        iSNS Server instance.  This is the number of rows in
        the isnsRegEntityTable for this instance."
        ::= { isnsNumObjectsEntry 3 }

isnsNumPortals             OBJECT-TYPE
    SYNTAX                 Gauge32 ( 0 .. 4294967295 )
    MAX-ACCESS              read-only
    STATUS                  current
    DESCRIPTION
        "The current total number of Portals registered in iSNS.
        This is the number of rows in isnsRegPortalTable."
        ::= { isnsNumObjectsEntry 4 }

isnsNumPortalGroups        OBJECT-TYPE
    SYNTAX                 Gauge32 ( 0 .. 4294967295 )
    MAX-ACCESS              read-only
    STATUS                  current
    DESCRIPTION
        "The current total number of Portal Groups registered in
        iSNS.  This is the number of rows in isnsRegPgTable."
        ::= { isnsNumObjectsEntry 5 }

```

```

isnsNumIscsiNodes          OBJECT-TYPE
    SYNTAX                  Gauge32 ( 0 .. 4294967295 )
    MAX-ACCESS              read-only
    STATUS                  current
    DESCRIPTION
        "The current total number of iSCSI node entries registered
        in the iSNS. This is the number rows in
        isnsRegIscsiNodeTable."
        ::= { isnsNumObjectsEntry 6 }

isnsNumFcPorts             OBJECT-TYPE
    SYNTAX                  Gauge32 ( 0 .. 4294967295 )
    MAX-ACCESS              read-only
    STATUS                  current
    DESCRIPTION
        "The current total number of FC Port entries registered
        in the iSNS. This is the number of rows in
        isnsRegFcPortTable."
        ::= { isnsNumObjectsEntry 7 }

isnsNumFcNodes             OBJECT-TYPE
    SYNTAX                  Gauge32 ( 0 .. 4294967295 )
    MAX-ACCESS              read-only
    STATUS                  current
    DESCRIPTION
        "The current total number of FC node entries registered
        in the iSNS. This is the number of rows in
        isnsRegFcNodeTable."
        ::= { isnsNumObjectsEntry 8 }

--
-- Control node information
--

isnsControlNodeInfo        OBJECT IDENTIFIER ::=
                                { isnsServerInfo 3 }

--
-- Specific iSCSI Nodes authorized to register as Control
-- Nodes
--

isnsControlNodeIscsiTable  OBJECT-TYPE
    SYNTAX                  SEQUENCE OF
                                IsnsControlNodeIscsiEntry
    MAX-ACCESS              not-accessible
    STATUS                  current
    DESCRIPTION

```

"Specified iSCSI Nodes that can register or are registered as control nodes. The number of rows is dependent on the number of iSCSI Control Nodes."

::= { isnsControlNodeInfo 1 }

isnsControlNodeIscsiEntry            OBJECT-TYPE  
     SYNTAX                            IsnsControlNodeIscsiEntry  
     MAX-ACCESS                        not-accessible  
     STATUS                            current  
     DESCRIPTION

"This is an iSCSI Control Node entry for a specific iSNS server instance."

    INDEX                            { isnsServerIndex,  
                                      isnsControlNodeIscsiNodeIndex }  
 ::= { isnsControlNodeIscsiTable 1 }

IsnsControlNodeIscsiEntry ::= SEQUENCE {  
     isnsControlNodeIscsiNodeIndex    IsnsNodeIndexId,  
     isnsControlNodeIscsiNodeName     SnmpAdminString,  
     isnsControlNodeIscsiIsRegistered TruthValue,  
     isnsControlNodeIscsiRcvMgtSCN    TruthValue  
     }

isnsControlNodeIscsiNodeIndex    OBJECT-TYPE  
     SYNTAX                            IsnsNodeIndexId  
     MAX-ACCESS                        not-accessible  
     STATUS                            current  
     DESCRIPTION

"The index for the iSCSI storage node authorized to act as a control node."

::= { isnsControlNodeIscsiEntry 1 }

isnsControlNodeIscsiNodeName    OBJECT-TYPE  
     SYNTAX                            SnmpAdminString  
     MAX-ACCESS                        read-only  
     STATUS                            current  
     DESCRIPTION

"The iSCSI Name of the initiator or target associated with the storage node. The iSCSI Name cannot be longer than 223 bytes. The iSNS Server internal maximum size is 224 bytes to provide NULL termination. This is the iSCSI Node Name for the storage node authorized and/or acting as a control node."

::= { isnsControlNodeIscsiEntry 2 }

isnsControlNodeIscsiIsRegistered OBJECT-TYPE  
     SYNTAX                            TruthValue  
     MAX-ACCESS                        read-only

```

        STATUS                current
        DESCRIPTION
        "Indicates whether the control node is currently
        registered in the iSNS Server instance."
        ::= { isnsControlNodeIscsiEntry 3 }

isnsControlNodeIscsiRcvMgtScn OBJECT-TYPE
    SYNTAX                TruthValue
    MAX-ACCESS            read-only
    STATUS                current
    DESCRIPTION
    "Indicates whether the Control Node has registered to
    receive Management SCNs. Management SCNs are sent to
    a Control Node if they are enabled, as indicated by
    isnsServerEnableControlNodeMgtScn, and the Control
    Node has registered for them."
    REFERENCE "RFC 4171, Section 2.2.3, 2.4"
    ::= { isnsControlNodeIscsiEntry 4 }

--
-- Specific FC Ports authorized to register as Control
-- Nodes
--

isnsControlNodeFcPortTable OBJECT-TYPE
    SYNTAX                SEQUENCE OF
                        IsnsControlNodeFcPortEntry
    MAX-ACCESS            not-accessible
    STATUS                current
    DESCRIPTION
    "Specified FC Ports that can register or are registered as
    control nodes. The number of rows is dependent on the
    number of FC Port Control Nodes."
    ::= { isnsControlNodeInfo 2 }

isnsControlNodeFcPortEntry OBJECT-TYPE
    SYNTAX                IsnsControlNodeFcPortEntry
    MAX-ACCESS            not-accessible
    STATUS                current
    DESCRIPTION
    "FC Port control node entry."
    INDEX                { isnsServerIndex,
                        isnsControlNodeFcPortWwpn }
    ::= { isnsControlNodeFcPortTable 1 }

IsnsControlNodeFcPortEntry ::= SEQUENCE {
    isnsControlNodeFcPortWwpn          FcNameIdOrZero,
    isnsControlNodeFcPortIsRegistered TruthValue,

```

```

        isnsControlNodeFcPortRcvMgtSCN      TruthValue
                                           }

isnsControlNodeFcPortWwpn  OBJECT-TYPE
    SYNTAX                  FcNameIdOrZero (SIZE(8))
    MAX-ACCESS              not-accessible
    STATUS                  current
    DESCRIPTION
        "The FC Port World Wide Port Name that can and/or is acting
        as a Control Node for the specified iSNS Server.  A zero-
        length string is not valid for this managed object.
        This managed object, combined with the isnsServerIndex, is
        the key for this table."
        ::= { isnsControlNodeFcPortEntry 1 }

isnsControlNodeFcPortIsRegistered OBJECT-TYPE
    SYNTAX                  TruthValue
    MAX-ACCESS              read-only
    STATUS                  current
    DESCRIPTION
        "Indicates whether the control node is currently
        registered in the iSNS Server instance."
        ::= { isnsControlNodeFcPortEntry 2 }

isnsControlNodeFcPortRcvMgtSCN OBJECT-TYPE
    SYNTAX                  TruthValue
    MAX-ACCESS              read-only
    STATUS                  current
    DESCRIPTION
        "Indicates whether the Control Node has registered to
        receive Management SCNs.  Management SCNs are sent to
        a Control Node if they are enabled, as indicated by
        isnsServerEnableControlNodeMgtScn, and the Control
        Node has registered for them."
        REFERENCE "RFC 4171, Section 2.2.3, 2.4"
        ::= { isnsControlNodeFcPortEntry 3 }

--
-- Discovery Domain Set information
--

isnsDdsInfo      OBJECT IDENTIFIER ::= { isnsServerInfo 4 }

--
-- Discovery Domain Set Registrations -----
--

isnsDdsTable      OBJECT-TYPE

```

```

SYNTAX                               SEQUENCE OF IsnsDdsEntry
MAX-ACCESS                           not-accessible
STATUS                               current
DESCRIPTION
"A table containing configuration information for each
Discovery Domain Set (DDS) registered in the iSNS Server
instance. The number of rows in the table is dependent
on the number of DDSs registered in the specified iSNS
server instance."
 ::= { isnsDdsInfo 1 }

isnsDdsEntry                          OBJECT-TYPE
SYNTAX                               IsnsDdsEntry
MAX-ACCESS                           not-accessible
STATUS                               current
DESCRIPTION
"Information on one Discovery Domain Set (DDS) registered
in the iSNS Server instance."
INDEX { isnsServerIndex, isnsDdsId }
 ::= { isnsDdsTable 1 }

IsnsDdsEntry ::=
SEQUENCE {
    isnsDdsId                      IsnsDiscoveryDomainSetId,
    isnsDdsSymbolicName            SnmpAdminString,
    isnsDdsStatus                  IsnsDdsStatusType
}

isnsDdsId                            OBJECT-TYPE
SYNTAX                               IsnsDiscoveryDomainSetId
MAX-ACCESS                           not-accessible
STATUS                               current
DESCRIPTION
"The ID that refers to this Discovery Domain Set and
index to the table."
 ::= { isnsDdsEntry 1 }

isnsDdsSymbolicName                  OBJECT-TYPE
SYNTAX                               SnmpAdminString
MAX-ACCESS                           read-only
STATUS                               current
DESCRIPTION
"The Discovery Domain Set Symbolic Name field contains
a unique variable-length description (up to 255 bytes)
that is associated with the DDS. If a Symbolic Name is
not provided, then one will be generated by the iSNS
server."
REFERENCE "RFC 4171, Section 6"

```

```

 ::= { isnsDdsEntry 2 }

isnsDdsStatus          OBJECT-TYPE
    SYNTAX              IsnsDdsStatusType
    MAX-ACCESS          read-only
    STATUS              current
    DESCRIPTION
        "The status of this Discovery Domain Set (DDS)."
```

REFERENCE "RFC 4171, Section 6.11.1.3"

```

 ::= { isnsDdsEntry 3 }

--
-- Discovery Domain Set Members -----
--

--
-- DDS Membership Assignment
--

isnsDdsMemberTable      OBJECT-TYPE
    SYNTAX              SEQUENCE OF IsnsDdsMemberEntry
    MAX-ACCESS          not-accessible
    STATUS              current
    DESCRIPTION
        "A table containing Discovery Domains (DDs) that have
        been assigned to specific Discovery Domain Sets (DDSs).
        The number of rows in the table is dependent on the
        number of DD to DDS relationships in the iSNS instance."
    ::= { isnsDdsInfo 2 }

isnsDdsMemberEntry      OBJECT-TYPE
    SYNTAX              IsnsDdsMemberEntry
    MAX-ACCESS          not-accessible
    STATUS              current
    DESCRIPTION
        "The mapping of one Discovery Domain (DD) to a Discovery
        Domain Set (DDS). This indicates the DD is a member of
        the DDS."
    INDEX { isnsServerIndex,
            isnsDdsId,
            isnsDdsMemberDdId }
    ::= { isnsDdsMemberTable 1 }

IsnsDdsMemberEntry ::=
    SEQUENCE {
        isnsDdsMemberDdId          IsnsDiscoveryDomainId,
        isnsDdsMemberSymbolicName SnmpAdminString
    }
```



```

    }

isnsDdsMemberDdId          OBJECT-TYPE
    SYNTAX                  IsnsDiscoveryDomainId
    MAX-ACCESS              not-accessible
    STATUS                  current
    DESCRIPTION
        "The ID that identifies the Discovery Domain
        that is a member of the Discovery Domain Set."
        ::= { isnsDdsMemberEntry 1 }

isnsDdsMemberSymbolicName  OBJECT-TYPE
    SYNTAX                  SnmpAdminString
    MAX-ACCESS              read-only
    STATUS                  current
    DESCRIPTION
        "The Symbolic Name of the Discovery Domain that is a member
        of this DDS. This value SHALL be identical to the object
        isnsDdSymbolicName for the associated DD ID."
        REFERENCE "RFC 4171, Section 6"
        ::= { isnsDdsMemberEntry 2 }

--
-- Discovery Domain information
--

isnsDdInfo          OBJECT IDENTIFIER ::= { isnsServerInfo 5 }

--
-- Discovery Domain Registrations -----
--

isnsDdTable          OBJECT-TYPE
    SYNTAX              SEQUENCE OF IsnsDdEntry
    MAX-ACCESS          not-accessible
    STATUS              current
    DESCRIPTION
        "A table containing configuration information for each
        Discovery Domain (DD) registered in the iSNS. The number
        of rows in the table is dependent on the number of DDS
        registered in the iSNS instance."
        ::= { isnsDdInfo 1 }

isnsDdEntry          OBJECT-TYPE
    SYNTAX              IsnsDdEntry
    MAX-ACCESS          not-accessible
    STATUS              current
    DESCRIPTION

```

"Information on a Discovery Domain (DD) registered in the iSNS Server instance."

```
INDEX    { isnsServerIndex, isnsDdId }
 ::= { isnsDdTable 1 }
```

IsnsDdEntry ::=

```
SEQUENCE {
    isnsDdId                IsnsDiscoveryDomainId,
    isnsDdSymbolicName      SnmpAdminString,
    isnsDdFeatures          IsnsDdFeatureType
}
```

```
isnsDdId                OBJECT-TYPE
    SYNTAX                IsnsDiscoveryDomainId
    MAX-ACCESS             not-accessible
    STATUS                 current
    DESCRIPTION
```

"The ID that refers to this Discovery Domain, and the index to the table."

```
REFERENCE "RFC 4171, Section 6"
 ::= { isnsDdEntry 1 }
```

```
isnsDdSymbolicName      OBJECT-TYPE
    SYNTAX                SnmpAdminString
    MAX-ACCESS             read-only
    STATUS                 current
    DESCRIPTION
```

"The Discovery Domain Symbolic Name field contains a unique variable-length description (up to 255 bytes) that is associated with the DD."

```
REFERENCE "RFC 4171, Section 6"
 ::= { isnsDdEntry 2 }
```

```
isnsDdFeatures          OBJECT-TYPE
    SYNTAX                IsnsDdFeatureType
    MAX-ACCESS             read-only
    STATUS                 current
    DESCRIPTION
```

"This defines the features the Discovery Domain has."

```
REFERENCE "RFC 4171, Section 6.11.2.9"
 ::= { isnsDdEntry 3 }
```

```
--
-- Discovery Domain Members -----
--

--
-- DD iSCSI Node Membership Assignment
--

isnsDdIscsiMemberTable      OBJECT-TYPE
    SYNTAX                  SEQUENCE OF
                            IsnsDdIscsiMemberEntry
    MAX-ACCESS               not-accessible
    STATUS                   current
    DESCRIPTION
        "A table containing iSCSI node indexes that have been
        assigned to specific DDs in this iSNS Server instance.  The
        number of rows in the table is dependent on the number of
        relationships between iSCSI Nodes and DDs registered in the
        iSNS instance."
        ::= { isnsDdInfo 2 }

isnsDdIscsiMemberEntry      OBJECT-TYPE
    SYNTAX                  IsnsDdIscsiMemberEntry
    MAX-ACCESS               not-accessible
    STATUS                   current
    DESCRIPTION
        "The mapping of one iSCSI Node to a Discovery Domain to
        indicate membership in the DD.  The indexes are the iSNS
        server instance, the DD ID of the Discovery Domain, and
        the iSCSI Node Index of the iSCSI Node."
    INDEX { isnsServerIndex,
            isnsDdId,
            isnsDdIscsiMemberIndex }
    ::= { isnsDdIscsiMemberTable 1 }

IsnsDdIscsiMemberEntry ::=
    SEQUENCE {
        isnsDdIscsiMemberIndex    IsnsNodeIndexId,
        isnsDdIscsiMemberName     SnmpAdminString,
        isnsDdIscsiMemberIsRegistered TruthValue
    }

isnsDdIscsiMemberIndex      OBJECT-TYPE
    SYNTAX                  IsnsNodeIndexId
    MAX-ACCESS               not-accessible
    STATUS                   current
    DESCRIPTION
        "The index for this member iSCSI node entry."
```

REFERENCE "RFC 4171, Section 6"  
 ::= { isnsDdIscsiMemberEntry 1 }

isnsDdIscsiMemberName OBJECT-TYPE  
 SYNTAX SnmpAdminString (SIZE (0..223))  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION  
 "The iSCSI Name associated with the storage node. The iSCSI Name cannot be longer than 223 bytes. The iSNS server internal maximum size is 224 bytes to provide NULL termination. This is the iSCSI Name for the storage node that is a member of the DD. This value maps 1 to 1 to the isnsDdIscsiMemberIndex node index. The iSCSI Name field is too long to be easily used for an index directly. The node index used for a specific node name is only persistent across iSNS Server reinitializations for nodes that are in a Discovery Domain (DD) or are registered control nodes. This value is only required during row creation if the storage node is not yet registered in the iSNS Server instance. If the storage node is not yet registered, then the iSCSI Name MUST be provided with the iSCSI node index during row creation in order to create the 1-to-1 mapping."

REFERENCE "RFC 4171, Section 6"  
 ::= { isnsDdIscsiMemberEntry 2 }

isnsDdIscsiMemberIsRegistered OBJECT-TYPE  
 SYNTAX TruthValue  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION  
 "This indicates whether this member of the DD is currently registered in the iSNS Server instance. iSCSI Storage Node members do not need to be currently registered in order for their iSCSI Name and Index to be added to a DD."

REFERENCE "RFC 4171, Section 6.11"  
 ::= { isnsDdIscsiMemberEntry 3 }

--  
 -- DD Portal Membership Assignment  
 --

isnsDdPortalMemberTable OBJECT-TYPE  
 SYNTAX SEQUENCE OF  
     IsnsDdPortalMemberEntry  
 MAX-ACCESS not-accessible

STATUS current  
DESCRIPTION

"A table containing currently registered and unregistered portal objects that have been explicitly assigned to specific DDs. Explicit assignment of a portal to a DD is only done when a specific set of portals are preferred for use within a DD. Otherwise, for iSCSI, the Portal Group Object should be used for identifying which portals provide access to which storage nodes. The number of rows in the table is dependent on the number of explicit relationships between portals and DDs registered in the iSNS."

REFERENCE "RFC 4171, Section 6"

::= { isnsDdInfo 3 }

isnsDdPortalMemberEntry OBJECT-TYPE  
SYNTAX IsnsDdPortalMemberEntry  
MAX-ACCESS not-accessible  
STATUS current  
DESCRIPTION

"Each entry indicates an explicit addition of a portal to a discovery domain. The explicit addition of an entity portal to a discovery domain indicates the portal is preferred for access to nodes of the entity for this discovery domain. Registered Portal Group objects are used in iSCSI to indicate mapping of portals to nodes across all discovery domains. Portals that have been explicitly mapped to a discovery domain will be returned as part of a query that is scoped to that discovery domain. If no portal of an entity has been explicitly mapped to a discovery domain, then all portals of the entity that provide access to a storage node are returned as part of a query. The table indexes are the server instance, the DD ID of the Discovery Domain, and the Portal Index of the portal."

INDEX { isnsServerIndex,  
isnsDdId,  
isnsDdPortalMemberIndex }  
::= { isnsDdPortalMemberTable 1 }

IsnsDdPortalMemberEntry ::= SEQUENCE {  
isnsDdPortalMemberIndex IsnsPortalIndexId,  
isnsDdPortalMemberAddressType InetAddressType,  
isnsDdPortalMemberAddress InetAddress,  
isnsDdPortalMemberPortType IsnsPortalPortTypeId,  
isnsDdPortalMemberPort InetPortNumber,  
isnsDdPortalMemberIsRegistered TruthValue  
}

```

isnsDdPortalMemberIndex      OBJECT-TYPE
    SYNTAX                    IsnsPortalIndexId
    MAX-ACCESS                not-accessible
    STATUS                    current
    DESCRIPTION
        "The index for a portal explicitly contained in the discovery
        domain. This managed object, combined with isnsServerIndex
        and isnsDdId, is the key for this table."
    REFERENCE "RFC 4171, Section 6"
    ::= { isnsDdPortalMemberEntry 1 }

isnsDdPortalMemberAddressType OBJECT-TYPE
    SYNTAX                    InetAddressType
    MAX-ACCESS                read-only
    STATUS                    current
    DESCRIPTION
        "The type of Inet address in isnsDdPortalMemberAddress. If
        the address is specified, then it must be a valid unicast
        address and the value of this object must be ipv4(1),
        ipv6(2), ipv4z(3), or ipv6z(4); otherwise, the value
        of this object is unknown(0), and the value of
        isnsDdPortalMemberAddress is the zero-length string."
    ::= { isnsDdPortalMemberEntry 2 }

isnsDdPortalMemberAddress     OBJECT-TYPE
    SYNTAX                    InetAddress
    MAX-ACCESS                read-only
    STATUS                    current
    DESCRIPTION
        "The Inet Address for the portal. The format of this
        object is specified by isnsDdPortalMemberAddressType."
    REFERENCE "RFC 4171, Section 6"
    ::= { isnsDdPortalMemberEntry 3 }

isnsDdPortalMemberPortType    OBJECT-TYPE
    SYNTAX                    IsnsPortalPortTypeId
    MAX-ACCESS                read-only
    STATUS                    current
    DESCRIPTION
        "The port type for the portal, either UDP or TCP."
    REFERENCE "RFC 4171, Section 6"
    ::= { isnsDdPortalMemberEntry 4 }

isnsDdPortalMemberPort        OBJECT-TYPE
    SYNTAX                    InetPortNumber ( 1 .. 65535 )
    MAX-ACCESS                read-only
    STATUS                    current

```

## DESCRIPTION

"The port number for the portal. Whether the portal type is TCP or UDP is indicated by isnsDdPortalMemberPortType."

REFERENCE "RFC 4171, Section 6"

::= { isnsDdPortalMemberEntry 5 }

isnsDdPortalMemberIsRegistered OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"This indicates whether this member of the DD is currently registered in the iSNS Server instance. Portals that are DD members do not need to be currently registered in order for them to be added to a DD."

REFERENCE "RFC 4171, Section 6.11"

::= { isnsDdPortalMemberEntry 6 }

--

-- DD FC Port Membership Assignment

--

isnsDdFcPortMemberTable OBJECT-TYPE

SYNTAX SEQUENCE OF

IsnsDdFcPortMemberEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

"A table containing FC Port World Wide Names (WWN) that have been assigned to specific DDs. The number of rows in the table is dependent on the number of relationships between FC Ports and DDs registered in the iSNS."

::= { isnsDdInfo 4 }

isnsDdFcPortMemberEntry OBJECT-TYPE

SYNTAX IsnsDdFcPortMemberEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

"The association of one FC Port with a Discovery Domain. Membership of an FC Port in a Discovery Domain is indicated by creating a row for the appropriate DD ID and FC Port WWN."

INDEX { isnsServerIndex,  
isnsDdId,  
isnsDdFcPortMemberPortName }

::= { isnsDdFcPortMemberTable 1 }

```

IsnsDdFcPortMemberEntry ::=
    SEQUENCE {
        isnsDdFcPortMemberPortName FcNameIdOrZero,
        isnsDdFcPortMemberIsRegistered TruthValue
    }

isnsDdFcPortMemberPortName OBJECT-TYPE
    SYNTAX          FcNameIdOrZero (SIZE(8))
    MAX-ACCESS       not-accessible
    STATUS           current
    DESCRIPTION
        "The Port WWN of the FC Port that is a member of the DD. The
        value MUST be a valid FC WWN, as per the FC-GS (Fibre Channel -
        Generic Services) standard. This managed object, combined
        with the isnsServerIndex and isnsDdId are the key for this
        table. A zero-length string is not a valid value for this
        managed object."
        REFERENCE "RFC 4171, Section 6"
        ::= { isnsDdFcPortMemberEntry 1 }

isnsDdFcPortMemberIsRegistered OBJECT-TYPE
    SYNTAX          TruthValue
    MAX-ACCESS       read-only
    STATUS           current
    DESCRIPTION
        "This indicates whether this member of the DD is currently
        registered in the iSNS Server instance."
        REFERENCE "RFC 4171, Section 6.11"
        ::= { isnsDdFcPortMemberEntry 2 }

--
-- Registered Device Information
--

isnsReg      OBJECT IDENTIFIER ::= { isnsServerInfo 6 }

isnsRegEntityInfo      OBJECT IDENTIFIER
                        ::= { isnsReg      1 }

--
-- iSNS Registered Entities Table
--

isnsRegEntityTable      OBJECT-TYPE
    SYNTAX          SEQUENCE OF IsnsRegEntityEntry
    MAX-ACCESS       not-accessible
    STATUS           current

```



## DESCRIPTION

"A table containing registered Entity objects in each iSNS server instance. The number of entries in the table is dependent on the number of Entity objects registered in the iSNS Server instances. All Entity objects are registered in the iSNS using the iSNS protocol."

```
::= { isnsRegEntityInfo 1 }
```

```
isnsRegEntityEntry      OBJECT-TYPE
    SYNTAX               IsnsRegEntityEntry
    MAX-ACCESS            not-accessible
    STATUS                current
```

## DESCRIPTION

"Information on one registered Entity object in an iSNS server instance."

```
INDEX      { isnsServerIndex,
              isnsRegEntityIndex }
 ::= { isnsRegEntityTable 1 }
```

```
IsnsRegEntityEntry ::=
    SEQUENCE {
        isnsRegEntityIndex      IsnsEntityIndexIdOrZero,
        isnsRegEntityEID        SnmpAdminString,
        isnsRegEntityProtocol    Unsigned32,
        isnsRegEntityManagementAddressType
                                InetAddressType,
        isnsRegEntityManagementAddress
                                InetAddress,
        isnsRegEntityTimestamp   TimeStamp,
        isnsRegEntityVersionMin   Unsigned32,
        isnsRegEntityVersionMax   Unsigned32,
        isnsRegEntityRegistrationPeriod
                                Unsigned32
    }
```

```
isnsRegEntityIndex      OBJECT-TYPE
    SYNTAX               IsnsEntityIndexIdOrZero
                        ( 1 .. 4294967295 )
    MAX-ACCESS            not-accessible
    STATUS                current
```

## DESCRIPTION

"The Entity Index for this entity. This index is assigned by the iSNS Server when an Entity is initially registered. The Entity Index can be used to represent a registered Entity object in situations where the Entity EID would be too long/unwieldy. Zero is not a valid value for this object."

REFERENCE "RFC 4171, Section 6"

```
::= { isnsRegEntityEntry 1 }
```

```
isnsRegEntityEID          OBJECT-TYPE
    SYNTAX                 SnmpAdminString
    MAX-ACCESS              read-only
    STATUS                  current
    DESCRIPTION
```

"The EID is a unique registered Entity object identifier, as specified in the iSNS Specification. This is the iSNS Entity Identifier for the registered Entity object."

```
    REFERENCE "RFC 4171, Section 6"
```

```
::= { isnsRegEntityEntry 2 }
```

```
isnsRegEntityProtocol     OBJECT-TYPE
    SYNTAX                 Unsigned32 ( 1 .. 4294967295 )
    MAX-ACCESS              read-only
    STATUS                  current
    DESCRIPTION
```

"The block storage protocol supported by this entity, as defined in the iSNS Specification, Section 6.2.2. The following values are initially assigned.

Type Value	Entity Type
-----	-----
1	No Protocol
2	iSCSI
3	iFCP
All Others	As assigned by IANA

The full set of current Block Storage Protocols are specified in the IANA-maintained registry of assigned iSNS parameters. Please refer to RFC 4171 and the iSNS parameters maintained at IANA."

```
    REFERENCE "RFC 4171, Section 6.2.2, and IANA Assignments"
```

```
::= { isnsRegEntityEntry 3 }
```

```
isnsRegEntityManagementAddressType OBJECT-TYPE
    SYNTAX                 InetAddressType
    MAX-ACCESS              read-only
    STATUS                  current
    DESCRIPTION
```

"The type of Inet address in isnsRegEntityManagementAddress. If the address is specified, then it must be a valid unicast address and the value of this object must be ipv4(1), ipv6(2), ipv4z(3), or ipv6z(4); otherwise, the value of this object is unknown(0), and the value of isnsRegEntityManagementAddress is the zero-length string."

```
::= { isnsRegEntityEntry 4 }
```

**isnsRegEntityManagementAddress OBJECT-TYPE**

SYNTAX                   InetAddress  
 MAX-ACCESS               read-only  
 STATUS                   current

**DESCRIPTION**

"The iSNS Management IP Address for the registered Entity object. The format of this object is specified by isnsRegEntityManagementAddressType."

REFERENCE "RFC 4171, Section 6"

::= { isnsRegEntityEntry 5 }

**isnsRegEntityTimestamp OBJECT-TYPE**

SYNTAX                   TimeStamp  
 MAX-ACCESS               read-only  
 STATUS                   current

**DESCRIPTION**

"The iSNS Entity Registration Timestamp for the registered Entity object. This is the most recent date and time that the registered Entity object, and associated registered objects contained in the Entity, were registered or updated."

REFERENCE "RFC 4171, Section 6"

::= { isnsRegEntityEntry 6 }

**isnsRegEntityVersionMin OBJECT-TYPE**

SYNTAX                   Unsigned32 ( 0 .. 254 | 255 )  
 MAX-ACCESS               read-only  
 STATUS                   current

**DESCRIPTION**

"The minimum version supported for the block storage protocol specified by isnsRegEntityProtocol. The protocol version specified can be from 1 to 254. A value of 255 is a wildcard value, indicating no minimum version value has been specified for this Entity. Entity registrations with an isnsRegEntityProtocol of 'No Protocol' SHALL have an isnsRegEntityVersionMin value of 0."

REFERENCE "RFC 4171, Section 6.2.5"

::= { isnsRegEntityEntry 7 }

**isnsRegEntityVersionMax OBJECT-TYPE**

SYNTAX                   Unsigned32 ( 0 .. 254 | 255 )  
 MAX-ACCESS               read-only  
 STATUS                   current

**DESCRIPTION**

"The maximum version supported for the block storage protocol specified by isnsRegEntityProtocol. The protocol version specified can be from 1 to 254. A value of 255 is a wildcard

value, indicating no maximum version value has been specified for this Entity. Entity registrations with an isnsRegEntityProtocol of 'No Protocol' SHALL have an isnsRegEntityVersionMax value of 0."

REFERENCE "RFC 4171, Section 6.2.5"

::= { isnsRegEntityEntry 8 }

isnsRegEntityRegistrationPeriod OBJECT-TYPE

SYNTAX Unsigned32 ( 0 .. 4294967295 )

UNITS "seconds"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The iSNS Entity Status Inquiry (ESI) registration period, which indicates the maximum time, in seconds, that the registration will be maintained without receipt of an iSNSP message from the entity. If the Registration Period is set to 0, then the Entity SHALL NOT be deregistered due to no contact with the entity."

REFERENCE "RFC 4171, Section 6"

::= { isnsRegEntityEntry 9 }

--

-- Registered Objects Associated With an Entity Information

--

isnsRegEntityNumObjectsTable OBJECT-TYPE

SYNTAX SEQUENCE OF

IsnsRegEntityNumObjectsEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A table containing information on the number of registered objects associated with a registered Entity in the iSNS server instance. The number of entries in the table is dependent on the number of registered Entity objects in the iSNS."

::= { isnsRegEntityInfo 2 }

isnsRegEntityNumObjectsEntry OBJECT-TYPE

SYNTAX IsnsRegEntityNumObjectsEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Information on the number of registered objects associated with a registered Entity object in an iSNS Server instance."

INDEX { isnsServerIndex,  
isnsRegEntityIndex }

```

 ::= { isnsRegEntityNumObjectsTable 1 }

IsnsRegEntityNumObjectsEntry ::=
    SEQUENCE {
        isnsRegEntityInfoNumPortals          Gauge32,
        isnsRegEntityInfoNumPortalGroups     Gauge32,
        isnsRegEntityInfoNumIscsiNodes       Gauge32,
        isnsRegEntityInfoNumFcPorts          Gauge32,
        isnsRegEntityInfoNumFcNodes          Gauge32
    }

isnsRegEntityInfoNumPortals OBJECT-TYPE
    SYNTAX      Gauge32 ( 0 .. 4294967295 )
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
    "The number of Portals associated with this Entity."
    ::= { isnsRegEntityNumObjectsEntry 1 }

isnsRegEntityInfoNumPortalGroups OBJECT-TYPE
    SYNTAX      Gauge32 ( 0 .. 4294967295 )
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
    "The number of Portal Groups associated with this Entity."
    ::= { isnsRegEntityNumObjectsEntry 2 }

isnsRegEntityInfoNumIscsiNodes OBJECT-TYPE
    SYNTAX      Gauge32 ( 0 .. 4294967295 )
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
    "The number of iSCSI Storage Nodes associated with this
    Entity."
    ::= { isnsRegEntityNumObjectsEntry 3 }

isnsRegEntityInfoNumFcPorts OBJECT-TYPE
    SYNTAX      Gauge32 ( 0 .. 4294967295 )
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
    "The number of FC Ports associated with this Entity."
    ::= { isnsRegEntityNumObjectsEntry 4 }

isnsRegEntityInfoNumFcNodes OBJECT-TYPE
    SYNTAX      Gauge32 ( 0 .. 4294967295 )
    MAX-ACCESS   read-only
    STATUS       current

```

```

        DESCRIPTION
        "The number of FC Nodes associated with this Entity."
        ::= { isnsRegEntityNumObjectsEntry 5 }

--
-- iSNS Registered Portal Information
--

isnsRegPortalInfo          OBJECT IDENTIFIER
                           ::= { isnsReg 2 }

--
-- iSNS Registered Portal Table
--

isnsRegPortalTable        OBJECT-TYPE
    SYNTAX                 SEQUENCE OF IsnsRegPortalEntry
    MAX-ACCESS              not-accessible
    STATUS                  current
    DESCRIPTION
        "A table containing the registered Portals in the iSNS.
        The number of entries is dependent on the number of
        Portals registered in the iSNS."
        ::= { isnsRegPortalInfo 1 }

isnsRegPortalEntry        OBJECT-TYPE
    SYNTAX                 IsnsRegPortalEntry
    MAX-ACCESS              not-accessible
    STATUS                  current
    DESCRIPTION
        "Information on one registered Entity Portal in the iSNS.
        The Entity Index is part of the table index to quickly
        find Portals that support a specific Entity."
    INDEX { isnsServerIndex,
            isnsRegEntityIndex,
            isnsRegPortalPortalIndex }
    ::= { isnsRegPortalTable 1 }

IsnsRegPortalEntry ::=
    SEQUENCE {
        isnsRegPortalPortalIndex    IsnsPortalIndexId,
        isnsRegPortalAddressType    InetAddressType,
        isnsRegPortalAddress        InetAddress,
        isnsRegPortalPortType       IsnsPortalPortTypeId,
        isnsRegPortalPort           InetPortNumber,
        isnsRegPortalSymbolicName   SnmpAdminString,
        isnsRegPortalEsiInterval    Unsigned32,
        isnsRegPortalEsiPortType    IsnsPortalPortTypeId,

```

```

        isnsRegPortalEsiPort      InetPortNumber,
        isnsRegPortalScnPortType  IsnsPortalPortTypeId,
        isnsRegPortalScnPort      InetPortNumber,
        isnsRegPortalSecurityInfo IsnsPortalSecurityType
    }

isnsRegPortalPortalIndex  OBJECT-TYPE
    SYNTAX                 IsnsPortalIndexId
    MAX-ACCESS              not-accessible
    STATUS                  current
    DESCRIPTION
        "The index for this Entity Portal."
    REFERENCE "RFC 4171, Section 6"
    ::= { isnsRegPortalEntry 1 }

isnsRegPortalAddressType  OBJECT-TYPE
    SYNTAX                 InetAddressType
    MAX-ACCESS              read-only
    STATUS                  current
    DESCRIPTION
        "The type of Inet address in isnsRegPortalAddress.  If the
        address is specified, then it must be a valid unicast
        address and the value of this object must be ipv4(1),
        ipv6(2), ipv4z(3), or ipv6z(4); otherwise, the value
        of this object is unknown(0), and the value of
        isnsRegPortalAddress is the zero-length string."
    ::= { isnsRegPortalEntry 2 }

isnsRegPortalAddress      OBJECT-TYPE
    SYNTAX                 InetAddress
    MAX-ACCESS              read-only
    STATUS                  current
    DESCRIPTION
        "The Inet Address for this Portal as defined in the iSNS
        Specification, RFC 4171.  The format of this object is
        specified by isnsRegPortalAddressType."
    REFERENCE "RFC 4171, Section 6"
    ::= { isnsRegPortalEntry 3 }

isnsRegPortalPortType     OBJECT-TYPE
    SYNTAX                 IsnsPortalPortTypeId
    MAX-ACCESS              read-only
    STATUS                  current
    DESCRIPTION
        "The port type for this Portal, either UDP or TCP, as
        defined in the iSNS Specification, RFC 4171."
    REFERENCE "RFC 4171, Section 6"
    ::= { isnsRegPortalEntry 4 }

```

```

isnsRegPortalPort          OBJECT-TYPE
    SYNTAX                  InetPortNumber ( 1 .. 65535 )
    MAX-ACCESS              read-only
    STATUS                  current
    DESCRIPTION
        "The port number for this Portal as defined in the
        iSNS Specification, RFC 4171.  Whether the Portal type
        is TCP or UDP is indicated by isnsRegPortalPortType."
    REFERENCE "RFC 4171, Section 6"
    ::= { isnsRegPortalEntry 5 }

isnsRegPortalSymbolicName  OBJECT-TYPE
    SYNTAX                  SnmpAdminString
    MAX-ACCESS              read-only
    STATUS                  current
    DESCRIPTION
        "The Symbolic Name for this Portal as defined in the iSNS
        Specification, RFC 4171.  If not provided, then the string
        SHALL be zero-length."
    REFERENCE "RFC 4171, Section 6"
    ::= { isnsRegPortalEntry 6 }

isnsRegPortalEsiInterval  OBJECT-TYPE
    SYNTAX                  Unsigned32 ( 0 .. 65535 )
    UNITS                    "seconds"
    MAX-ACCESS              read-only
    STATUS                  current
    DESCRIPTION
        "The Entity Status Inquiry (ESI) Interval for this Portal
        as defined in the iSNS Specification, RFC 4171.  A value of
        0 indicates that ESI monitoring has not been configured for
        this Portal."
    REFERENCE "RFC 4171, Section 6.3.4"
    ::= { isnsRegPortalEntry 7 }

isnsRegPortalEsiPortType  OBJECT-TYPE
    SYNTAX                  IsnsPortalPortTypeId
    MAX-ACCESS              read-only
    STATUS                  current
    DESCRIPTION
        "The port type for the ESI Port, either UDP or TCP, as
        defined in the iSNS Specification, RFC 4171."
    REFERENCE "RFC 4171, Section 6"
    ::= { isnsRegPortalEntry 8 }

isnsRegPortalEsiPort      OBJECT-TYPE
    SYNTAX                  InetPortNumber

```



```

MAX-ACCESS          read-only
STATUS              current
DESCRIPTION
"The TCP or UDP port number used for ESI monitoring.  Whether
the port type is TCP or UDP is indicated by
isnsRegPortalEsiPortType.  A value of 0 indicates that ESI
monitoring is not enabled for this Portal."
REFERENCE "RFC 4171, Section 6"
 ::= { isnsRegPortalEntry 9 }

isnsRegPortalScnPortType OBJECT-TYPE
SYNTAX              IsnsPortalPortTypeId
MAX-ACCESS          read-only
STATUS              current
DESCRIPTION
"The port type for the SCN Port, either UDP or TCP, as
defined in the iSNS Specification, RFC 4171."
REFERENCE "RFC 4171, Section 6"
 ::= { isnsRegPortalEntry 10 }

isnsRegPortalScnPort OBJECT-TYPE
SYNTAX              InetPortNumber
MAX-ACCESS          read-only
STATUS              current
DESCRIPTION
"The TCP or UDP port used to receive SCN messages from the
iSNS Server.  Whether the port type is TCP or UDP is
indicated by isnsRegPortalScnPortType.  A value of 0
indicates that SCN message receipt is not enabled for this
Portal."
REFERENCE "RFC 4171, Section 6"
 ::= { isnsRegPortalEntry 11 }

isnsRegPortalSecurityInfo OBJECT-TYPE
SYNTAX              IsnsPortalSecurityType
MAX-ACCESS          read-only
STATUS              current
DESCRIPTION
"Indicates security attribute settings for the Portal as
registered in the iSNS server.  The bit for bitmapVALID must
be set in order for this attribute to contain valid
information.  Setting a bit to 1 indicates the
feature is enabled."
REFERENCE "RFC 4171, Section 6.3.9"
 ::= { isnsRegPortalEntry 12 }

```

```

--
-- iSNS Registered Portal Group Information
--

isnsRegPortalGroupInfo      OBJECT IDENTIFIER
                             ::= { isnsReg 3 }

--
-- iSNS Registered Portal Group (PG) Table
--

isnsRegPgTable              OBJECT-TYPE
    SYNTAX                   SEQUENCE OF IsnsRegPgEntry
    MAX-ACCESS                not-accessible
    STATUS                    current
    DESCRIPTION
        "A table containing the registered Portal Groups (PGs) in
        the iSNS Server instance. The number of entries is
        dependent on the number of Portal Groups registered in
        the iSNS."
    ::= { isnsRegPortalGroupInfo 1 }

isnsRegPgEntry              OBJECT-TYPE
    SYNTAX                   IsnsRegPgEntry
    MAX-ACCESS                not-accessible
    STATUS                    current
    DESCRIPTION
        "Information on one registered Portal Group in the iSNS
        server instance. The Entity Index is part of the table
        index to quickly find Portal Groups that support Portals
        and iSCSI Storage Nodes in a specific Entity."
    INDEX { isnsServerIndex,
            isnsRegEntityIndex,
            isnsRegPgIndex }
    ::= { isnsRegPgTable 1 }

IsnsRegPgEntry ::=
    SEQUENCE {
        isnsRegPgIndex          IsnsPortalGroupIndexId,
        isnsRegPgIscsiNodeIndex IsnsNodeIndexId,
        isnsRegPgIscsiName      SnmpAdminString,
        isnsRegPgPortalPortalIndex IsnsPortalIndexId,
        isnsRegPgPortalAddressType InetAddressType,
        isnsRegPgPortalAddress   InetAddress,
        isnsRegPgPortalPortType  IsnsPortalPortTypeId,
        isnsRegPgPortalPort      InetPortNumber,
        isnsRegPgPGT             IsnsPortalGroupTagIdOrNull
    }

```

```

isnsRegPgIndex          OBJECT-TYPE
    SYNTAX               IsnsPortalGroupIndexId
    MAX-ACCESS            not-accessible
    STATUS                current
    DESCRIPTION
        "The PG Index for this node.  The index is created by the
        iSNS Server instance for uniquely identifying registered
        objects.  The PG object is registered at the same time a
        Portal or Storage Node is registered using the iSNS
        protocol."
        REFERENCE "RFC 4171, Section 6"
        ::= { isnsRegPgEntry 1 }

isnsRegPgIscsiNodeIndex OBJECT-TYPE
    SYNTAX               IsnsNodeIndexId
    MAX-ACCESS            read-only
    STATUS                current
    DESCRIPTION
        "The index for the iSCSI Node associated with this PG.
        This index can be used to reference the
        isnsRegIscsiNodeTable."
        REFERENCE "RFC 4171, Section 6"
        ::= { isnsRegPgEntry 2 }

isnsRegPgIscsiName       OBJECT-TYPE
    SYNTAX               SnmpAdminString (SIZE (0..223))
    MAX-ACCESS            read-only
    STATUS                current
    DESCRIPTION
        "The iSCSI Name of the initiator or target associated with
        the storage node.  The iSCSI Name cannot be longer than
        223 bytes.  The iSNS Server internal maximum size is 224
        bytes to provide NULL termination.  This is the PG iSCSI
        Name that uniquely identifies the iSCSI Storage Node that
        is associated with this PG."
        ::= { isnsRegPgEntry 3 }

isnsRegPgPortalPortalIndex OBJECT-TYPE
    SYNTAX               IsnsPortalIndexId
    MAX-ACCESS            read-only
    STATUS                current
    DESCRIPTION
        "The Portal Index for the Portal associated with this PG.
        This index can be used to reference the isnsRegPortalTable."
        ::= { isnsRegPgEntry 4 }

isnsRegPgPortalAddressType OBJECT-TYPE

```

SYNTAX	InetAddressType
MAX-ACCESS	read-only
STATUS	current
DESCRIPTION	
<p>"The type of Inet address in isnsRegPgPortalAddress. If the address is specified, then it must be a valid unicast address and the value of this object must be ipv4(1), ipv6(2), ipv4z(3), or ipv6z(4); otherwise, the value of this object is unknown(0), and the value of isnsRegPgPortalAddress is the zero-length string."</p> <p>::= { isnsRegPgEntry 5 }</p>	
isnsRegPgPortalAddress	OBJECT-TYPE
SYNTAX	InetAddress
MAX-ACCESS	read-only
STATUS	current
DESCRIPTION	
<p>"The Inet Address for the Portal that is associated with the PG. The format of this object is specified by isnsRegPgPortalAddressType."</p> <p>REFERENCE "RFC 4171, Section 6"</p> <p>::= { isnsRegPgEntry 6 }</p>	
isnsRegPgPortalPortType	OBJECT-TYPE
SYNTAX	IsnsPortalPortTypeId
MAX-ACCESS	read-only
STATUS	current
DESCRIPTION	
<p>"The port type, either UDP or TCP, for the Portal that is associated with this registered PG object."</p> <p>REFERENCE "RFC 4171, Section 6"</p> <p>::= { isnsRegPgEntry 7 }</p>	
isnsRegPgPortalPort	OBJECT-TYPE
SYNTAX	InetPortNumber ( 1 .. 65535 )
MAX-ACCESS	read-only
STATUS	current
DESCRIPTION	
<p>"The port number for the Portal that is associated with this registered PG object. Whether the Portal type is TCP or UDP is indicated by isnsRegPgPortalPortType."</p> <p>REFERENCE "RFC 4171, Section 6"</p> <p>::= { isnsRegPgEntry 8 }</p>	
isnsRegPgPGT	OBJECT-TYPE
SYNTAX	IsnsPortalGroupTagIdOrNull
MAX-ACCESS	read-only
STATUS	current

## DESCRIPTION

"The Portal Group Tag (PGT) for the registered iSCSI Portal Group object in an iSNS Server instance. This indicates the tag value that the Portal uses for access to the iSCSI Storage Node. The PGT is used for coordinated access between multiple Portals, as described in the iSCSI Specification, RFC 3720. A PGT with no association is a NULL value. The value of -1 indicates a NULL value."

REFERENCE "RFC 4171, Section 6, and RFC 3720"

::= { isnsRegPgEntry 9 }

--

-- iSNS Registered iSCSI Node Information

--

isnsRegIscsiNodeInfo OBJECT IDENTIFIER ::= { isnsReg 4 }

--

-- iSNS Registered iSCSI Node Table

--

isnsRegIscsiNodeTable	OBJECT-TYPE
SYNTAX	SEQUENCE OF IsnsRegIscsiNodeEntry
MAX-ACCESS	not-accessible
STATUS	current

## DESCRIPTION

"A table containing the registered iSCSI Nodes in the iSNS server instance. Storage devices register using the iSNS protocol. While a device cannot be registered in an iSNS server using SNMP, an entry can be deleted in order to remove 'stale' entries. The number of entries is related to the number of iSCSI nodes registered in the iSNS."

::= { isnsRegIscsiNodeInfo 1 }

isnsRegIscsiNodeEntry	OBJECT-TYPE
SYNTAX	IsnsRegIscsiNodeEntry
MAX-ACCESS	not-accessible
STATUS	current

## DESCRIPTION

"Information on one iSCSI node that has been registered in the iSNS Server instance. New rows cannot be added using SNMP."

INDEX { isnsServerIndex,  
          isnsRegEntityIndex,  
          isnsRegIscsiNodeIndex }

::= { isnsRegIscsiNodeTable 1 }

IsnsRegIscsiNodeEntry ::= SEQUENCE {

```

    isnsRegIscsiNodeIndex      IsnsNodeIndexId,
    isnsRegIscsiNodeName      SnmpAdminString,
    isnsRegIscsiNodeType      IsnsIscsiNodeType,
    isnsRegIscsiNodeAlias      SnmpAdminString,
    isnsRegIscsiNodeScnTypes   IsnsIscsiScnType,
    isnsRegIscsiNodeWwnToken   FcNameIdOrZero,
    isnsRegIscsiNodeAuthMethod SnmpAdminString
    }

isnsRegIscsiNodeIndex      OBJECT-TYPE
    SYNTAX                  IsnsNodeIndexId
    MAX-ACCESS               not-accessible
    STATUS                   current
    DESCRIPTION
        "The index for this iSCSI node."
    REFERENCE "RFC 4171, Section 6"
    ::= { isnsRegIscsiNodeEntry 1 }

isnsRegIscsiNodeName      OBJECT-TYPE
    SYNTAX                  SnmpAdminString (SIZE (0..223))
    MAX-ACCESS               read-only
    STATUS                   current
    DESCRIPTION
        "The iSCSI Name of the initiator or target associated with
        the storage node. The iSCSI Name cannot be longer than
        223 bytes. The iSNS Server internal maximum size is 224
        bytes to provide NULL termination. This is the iSCSI Name
        that uniquely identifies the initiator, initiator/target,
        target, or control node in the network."
    REFERENCE "RFC 4171, Section 6"
    ::= { isnsRegIscsiNodeEntry 2 }

isnsRegIscsiNodeType      OBJECT-TYPE
    SYNTAX                  IsnsIscsiNodeType
    MAX-ACCESS               read-only
    STATUS                   current
    DESCRIPTION
        "The Node Type defining the functions of this iSCSI node."
    ::= { isnsRegIscsiNodeEntry 3 }

isnsRegIscsiNodeAlias      OBJECT-TYPE
    SYNTAX                  SnmpAdminString
    MAX-ACCESS               read-only
    STATUS                   current
    DESCRIPTION
        "The Alias name of the iSCSI node. This is a variable-length
        text-based description of up to 255 bytes."
    REFERENCE "RFC 4171, Section 6"

```

```

 ::= { isnsRegIscsiNodeEntry 4 }

isnsRegIscsiNodeScnTypes      OBJECT-TYPE
    SYNTAX                    IsnsIscsiScnType
    MAX-ACCESS                read-only
    STATUS                    current
    DESCRIPTION
        "The State Change Notification (SCN) types enabled for this
        iSCSI node."
    REFERENCE "RFC 4171, Section 6.4.4"
    ::= { isnsRegIscsiNodeEntry 5 }

isnsRegIscsiNodeWwnToken      OBJECT-TYPE
    SYNTAX                    FcNameIdOrZero
    MAX-ACCESS                read-only
    STATUS                    current
    DESCRIPTION
        "This 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 RFC 4171. A zero-length string
        for this managed object indicates that a Node WWN token has
        not been assigned."
    REFERENCE "RFC 4171, Section 6"
    ::= { isnsRegIscsiNodeEntry 6 }

isnsRegIscsiNodeAuthMethod    OBJECT-TYPE
    SYNTAX                    SnmpAdminString
    MAX-ACCESS                read-only
    STATUS                    current
    DESCRIPTION
        "This attribute contains a null-terminated string containing
        UTF-8 text listing the iSCSI authentication methods enabled
        for this iSCSI 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
        RFC 3720 - iSCSI. Additional vendor-specific text values
        are also possible."
    REFERENCE "RFC 4171, Section 6, and RFC 3720"
    ::= { isnsRegIscsiNodeEntry 7 }

--
-- iSNS Registered FC Node Information
--

isnsRegFcNodeInfo             OBJECT IDENTIFIER ::= { isnsReg 5 }

```

```

--
-- iSNS Registered FC Node Table
--

isnsRegFcNodeTable          OBJECT-TYPE
    SYNTAX                   SEQUENCE OF IsnsRegFcNodeEntry
    MAX-ACCESS                not-accessible
    STATUS                    current
    DESCRIPTION
        "A table containing the registered FC Nodes in the iSNS.
        This supports iFCP as defined in RFC 4172."
        ::= { isnsRegFcNodeInfo 1 }

isnsRegFcNodeEntry          OBJECT-TYPE
    SYNTAX                   IsnsRegFcNodeEntry
    MAX-ACCESS                not-accessible
    STATUS                    current
    DESCRIPTION
        "Information on one registered FC node that has been
        registered in the iSNS."
        INDEX { isnsServerIndex,
                 isnsRegFcNodeWwnn }
        ::= { isnsRegFcNodeTable 1 }

IsnsRegFcNodeEntry ::= SEQUENCE {
    isnsRegFcNodeWwnn         FcNameIdOrZero,
    isnsRegFcNodeSymbolicName SnmpAdminString,
    isnsRegFcNodeAddressType  InetAddressType,
    isnsRegFcNodeAddress      InetAddress,
    isnsRegFcNodeIPA          OCTET STRING,
    isnsRegFcNodeProxyIscsiName SnmpAdminString,
    isnsRegFcNodeNumFcPorts   Gauge32
}

isnsRegFcNodeWwnn           OBJECT-TYPE
    SYNTAX                   FcNameIdOrZero (SIZE(8))
    MAX-ACCESS                not-accessible
    STATUS                    current
    DESCRIPTION
        "The FC Node World Wide Node Name as defined in the iSNS
        Specification, RFC 4171. A zero-length string is not valid
        for this managed object."
        REFERENCE "RFC 4171, Section 6"
        ::= { isnsRegFcNodeEntry 1 }

isnsRegFcNodeSymbolicName   OBJECT-TYPE
    SYNTAX                   SnmpAdminString

```



```

    MAX-ACCESS          read-only
    STATUS              current
    DESCRIPTION
    "The FC Node Symbolic Name of the node as defined in the
    iSNS Specification, RFC 4171. This is a variable-length
    text-based description. If not provided, then the string
    SHALL be zero-length."
    REFERENCE "RFC 4171, Section 6"
    ::= { isnsRegFcNodeEntry 2 }

isnsRegFcNodeAddressType OBJECT-TYPE
    SYNTAX              InetAddressType
    MAX-ACCESS          read-only
    STATUS              current
    DESCRIPTION
    "The type of Inet address in isnsRegFcNodeAddress. If
    the address is specified, then it must be a valid unicast
    address and the value of this object must be ipv4(1),
    ipv6(2), ipv4z(3), or ipv6z(4); otherwise, the value
    of this object is unknown(0), and the value of
    isnsRegFcNodeAddress is the zero-length string."
    ::= { isnsRegFcNodeEntry 3 }

isnsRegFcNodeAddress OBJECT-TYPE
    SYNTAX              InetAddress
    MAX-ACCESS          read-only
    STATUS              current
    DESCRIPTION
    "The FC Node Inet address of the node as defined in the
    iSNS Specification, RFC 4171. The format of this object is
    specified by isnsRegFcNodeAddressType."
    REFERENCE "RFC 4171, Section 6"
    ::= { isnsRegFcNodeEntry 4 }

isnsRegFcNodeIPA OBJECT-TYPE
    SYNTAX              OCTET STRING (SIZE(8))
    MAX-ACCESS          read-only
    STATUS              current
    DESCRIPTION
    "This managed object identifies the FC Initial Process
    Associator of the node as defined in the iSNS
    Specification, RFC 4171."
    REFERENCE "RFC 4171, Section 6"
    ::= { isnsRegFcNodeEntry 5 }

isnsRegFcNodeProxyIscsiName OBJECT-TYPE
    SYNTAX              SnmpAdminString (SIZE (0..223))
    MAX-ACCESS          read-only

```

```

        STATUS                                current
        DESCRIPTION
        "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."
        REFERENCE "RFC 4171, Section 6"
        ::= { isnsRegFcNodeEntry 6 }

isnsRegFcNodeNumFcPorts      OBJECT-TYPE
    SYNTAX                    Gauge32 ( 0 .. 4294967295 )
    MAX-ACCESS                 read-only
    STATUS                     current
    DESCRIPTION
    "The number of FC Ports associated with this FC Node."
    ::= { isnsRegFcNodeEntry 7 }

--
-- iSNS Registered FC Port Table
--

isnsRegFcPortTable           OBJECT-TYPE
    SYNTAX                    SEQUENCE OF IsnsRegFcPortEntry
    MAX-ACCESS                 not-accessible
    STATUS                     current
    DESCRIPTION
    "Information on registered FC N_Ports in the iSNS. FC Ports
    are associated with registered FC Nodes. This supports
    iFCP as defined in RFC 4172."
    REFERENCE "RFC 4172, Section 4"
    ::= { isnsRegFcNodeInfo 2 }

isnsRegFcPortEntry           OBJECT-TYPE
    SYNTAX                    IsnsRegFcPortEntry
    MAX-ACCESS                 not-accessible
    STATUS                     current
    DESCRIPTION
    "Information on one FC Port that has been registered in
    iSNS."
    REFERENCE "RFC 4172, Section 4"
    INDEX { isnsServerIndex,
            isnsRegEntityIndex,
            isnsRegFcPortWwpn }
    ::= { isnsRegFcPortTable 1 }

IsnsRegFcPortEntry ::= SEQUENCE {
    isnsRegFcPortWwpn          FcNameIdOrZero,

```

```

isnsRegFcPortID          FcAddressIdOrZero,
isnsRegFcPortType        Unsigned32,
isnsRegFcPortSymbolicName SnmpAdminString,
isnsRegFcPortFabricPortWwn FcNameIdOrZero,
isnsRegFcPortHA          FcAddressIdOrZero,
isnsRegFcPortAddressType InetAddressType,
isnsRegFcPortAddress     InetAddress,
isnsRegFcPortFcCos       IsnsFcClassOfServiceType,
isnsRegFcPortFc4Types    OCTET STRING,
isnsRegFcPortFc4Descr    SnmpAdminString,
isnsRegFcPortFc4Features OCTET STRING,
isnsRegFcPortScnTypes    IsnsIfcpScnType,
isnsRegFcPortRole        IsnsFcPortRoleType,
isnsRegFcPortFcNodeWwn   FcNameIdOrZero,
isnsRegFcPortPpnWwn      FcNameIdOrZero
    }

```

```

isnsRegFcPortWwpn      OBJECT-TYPE
    SYNTAX               FcNameIdOrZero (SIZE(8))
    MAX-ACCESS            not-accessible
    STATUS                current
    DESCRIPTION

```

"The FC Port's World Wide Port Name as defined in the iSNS Specification, RFC 4171. A zero-length string is not valid for this managed object."

```

REFERENCE "RFC 4171, Section 6"
 ::= { isnsRegFcPortEntry 1 }

```

```

isnsRegFcPortID        OBJECT-TYPE
    SYNTAX               FcAddressIdOrZero
    MAX-ACCESS            read-only
    STATUS                current
    DESCRIPTION

```

"The FC Port's Port ID as defined in the iSNS Specification, RFC 4171."

```

REFERENCE "RFC 4171, Section 6"
 ::= { isnsRegFcPortEntry 2 }

```

```

isnsRegFcPortType      OBJECT-TYPE
    SYNTAX               Unsigned32 ( 0 .. 65535 )
    MAX-ACCESS            read-only
    STATUS                current
    DESCRIPTION

```

"The FC Port Type as defined in the iSNS Specification, RFC 4171, and the Fibre Channel Generic Services Specification. Current values are as shown below:

```

        unknown      (0),
        nPort        (1),

```

```

nlPort      (2),
fNlPort     (3),
fPort       (129),      -- x'81'
flPort      (130),      -- x'82'
ePort       (132),      -- x'84'
bPort       (133),      -- x'85'
mFcpPort    (65297),    -- x'FF11'
iFcpPort    (65298),    -- x'FF12'
unknownEnd  (65535)

```

The future assignment of any additional values will be documented in a revision of RFC 4171."

REFERENCE "RFC 4171, Section 6.6.3"

::= { isnsRegFcPortEntry 3 }

```

isnsRegFcPortSymbolicName OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION

```

"The FC Port Symbolic Name as defined in the iSNS Specification, RFC 4171. If not provided, then the string SHALL be zero-length."

REFERENCE "RFC 4171, Section 6"

::= { isnsRegFcPortEntry 4 }

```

isnsRegFcPortFabricPortWwn OBJECT-TYPE
    SYNTAX      FcNameIdOrZero
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION

```

"The Fabric Port WWN for this entry as defined in the iSNS Specification, RFC 4171. A zero-length string for this managed object indicates that the Fabric Port WWN is not known, or has not yet been registered with the iSNS Server."

REFERENCE "RFC 4171, Section 6"

::= { isnsRegFcPortEntry 5 }

```

isnsRegFcPortHA OBJECT-TYPE
    SYNTAX      FcAddressIdOrZero
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION

```

"The FC Port Hard Address as defined in the iSNS Specification, RFC 4171."

REFERENCE "RFC 4171, Section 6"

::= { isnsRegFcPortEntry 6 }

```

isnsRegFcPortAddressType OBJECT-TYPE

```

```

SYNTAX                               InetAddressType
MAX-ACCESS                           read-only
STATUS                               current
DESCRIPTION
"The type of Inet address in isnsRegFcPortAddress.  If
the address is specified, then it must be a valid unicast
address and the value of this object must be ipv4(1),
ipv6(2), ipv4z(3), or ipv6z(4); otherwise, the value
of this object is unknown(0), and the value of
isnsRegFcPortAddress is the zero-length string."
 ::= { isnsRegFcPortEntry 7 }

isnsRegFcPortAddress                 OBJECT-TYPE
SYNTAX                               InetAddress
MAX-ACCESS                           read-only
STATUS                               current
DESCRIPTION
"The FC Port Inet Address as defined in the iSNS
Specification, RFC 4171.  The format of this object is
specified by isnsRegFcPortAddressType."
REFERENCE "RFC 4171, Section 6"
 ::= { isnsRegFcPortEntry 8 }

isnsRegFcPortFcCos                   OBJECT-TYPE
SYNTAX                               IsnsFcClassOfServiceType
MAX-ACCESS                           read-only
STATUS                               current
DESCRIPTION
"The FC Port Class of Service as defined in the iSNS
Specification, RFC 4171."
REFERENCE "RFC 4171, Section 6"
 ::= { isnsRegFcPortEntry 9 }

isnsRegFcPortFc4Types                OBJECT-TYPE
SYNTAX                               OCTET STRING (SIZE (32))
MAX-ACCESS                           read-only
STATUS                               current
DESCRIPTION
"The FC Port FC-4 Types as defined in the iSNS
Specification, RFC 4171."
REFERENCE "RFC 4171, Section 6.6.9"
 ::= { isnsRegFcPortEntry 10 }

isnsRegFcPortFc4Descr                OBJECT-TYPE
SYNTAX                               SnmpAdminString (SIZE(4..255))
MAX-ACCESS                           read-only
STATUS                               current
DESCRIPTION

```

"The FC Port FC-4 Descriptor as defined in the iSNS Specification, RFC 4171. The FC-4 Descriptor cannot be longer than 255 bytes. The iSNS Server internal maximum size is 256 bytes to provide NULL termination."

REFERENCE "RFC 4171, Section 6.6.10"

::= { isnsRegFcPortEntry 11 }

isnsRegFcPortFc4Features	OBJECT-TYPE
SYNTAX	OCTET STRING (SIZE (128))
MAX-ACCESS	read-only
STATUS	current
DESCRIPTION	

"The FC Port FC-4 Features as defined in the iSNS Specification, RFC 4171."

REFERENCE "RFC 4171, Section 6.6.11"

::= { isnsRegFcPortEntry 12 }

isnsRegFcPortScnTypes	OBJECT-TYPE
SYNTAX	IsnsIfcpScnType
MAX-ACCESS	read-only
STATUS	current
DESCRIPTION	

"The iFCP State Change Notification (SCN) types enabled for the registered object."

REFERENCE "RFC 4171, Section 6"

::= { isnsRegFcPortEntry 13 }

isnsRegFcPortRole	OBJECT-TYPE
SYNTAX	IsnsFcPortRoleType
MAX-ACCESS	read-only
STATUS	current
DESCRIPTION	

"The FC Port Role defines the role of the registered object."

REFERENCE "RFC 4171, Section 6"

::= { isnsRegFcPortEntry 14 }

isnsRegFcPortFcNodeWwnn	OBJECT-TYPE
SYNTAX	FcNameIdOrZero
MAX-ACCESS	read-only
STATUS	current
DESCRIPTION	

"The FC Node World Wide Node Name that is associated with this FC Port as defined in the iSNS Specification, RFC 4171. This managed object may contain a zero-length string prior to a device registering this value with the iSNS Server."

REFERENCE "RFC 4171, Section 6"

::= { isnsRegFcPortEntry 15 }

```

isnsRegFcPortPpnWwn          OBJECT-TYPE
    SYNTAX                    FcNameIdOrZero
    MAX-ACCESS                read-only
    STATUS                    current
    DESCRIPTION
        "The Permanent Port Name (PPN) attribute is the FC Port Name WWPN
        of the first Storage Node registered in the iSNS Database
        that is associated with a particular FC Device (FC Node).
        The PPN of all subsequent Storage Node registrations that
        are associated with that FC Device (FC Node) SHALL be set
        to the FC Port Name WWPN of the first Storage Node, as
        defined in the iSNS Specification, RFC 4171. This managed
        object may contain a zero-length string prior to a device
        registering this value with the iSNS Server."
    REFERENCE "RFC 4171, Section 6"
    ::= { isnsRegFcPortEntry 16 }

--
-- Mapping from FC Node to Entity - FC Port
--

isnsRegFcNodePortTable       OBJECT-TYPE
    SYNTAX                    SEQUENCE OF
                                IsnsRegFcNodePortEntry
    MAX-ACCESS                not-accessible
    STATUS                    current
    DESCRIPTION
        "A table containing the mapping of a registered FC Node and
        associated registered iFCP Port to the supporting registered
        Entity object in an iSNS Server instance."
    ::= { isnsRegFcNodeInfo 3 }

isnsRegFcNodePortEntry       OBJECT-TYPE
    SYNTAX                    IsnsRegFcNodePortEntry
    MAX-ACCESS                not-accessible
    STATUS                    current
    DESCRIPTION
        "Information on one mapping from an FC Node and iFCP Port to
        an Entity object registered in an iSNS."
    INDEX { isnsServerIndex,
            isnsRegFcNodeWwnn,
            isnsRegFcPortWwpn }
    ::= { isnsRegFcNodePortTable 1 }

IsnsRegFcNodePortEntry ::= SEQUENCE {
    isnsRegFcNodePortEntityIndex IsnsEntityIndexIdOrZero
}
```

```

isnsRegFcNodePortEntityIndex OBJECT-TYPE
    SYNTAX                IsnsEntityIndexIdOrZero
    MAX-ACCESS             read-only
    STATUS                 current
    DESCRIPTION
        "The Entity Index for the registered Entity object
        associated with the FC Port and FC Node. This managed
        object may contain the value of zero prior to a device
        registering this value with the iSNS Server."
    ::= { isnsRegFcNodePortEntry 1 }

--
-- iSNS Notifications Information -----
--

isnsNotificationsInfo      OBJECT IDENTIFIER
                            ::= { isnsObjects 2 }

isnsInstanceInfo           OBJECT-TYPE
    SYNTAX                 SnmpAdminString
    MAX-ACCESS             accessible-for-notify
    STATUS                 current
    DESCRIPTION
        "Textual information about the notification event and the
        iSNS Server generating the notification. An example is:
        iSNS Server Started."
    ::= { isnsNotificationsInfo 1 }

isnsAddressNotificationType OBJECT-TYPE
    SYNTAX                 InetAddressType
    MAX-ACCESS             accessible-for-notify
    STATUS                 current
    DESCRIPTION
        "The type of Inet address in isnsAddressNotification. If
        the address is specified, then it must be a valid unicast
        address and the value of this object must be ipv4(1),
        ipv6(2), ipv4z(3), or ipv6z(4); otherwise, the value
        of this object is unknown(0), and the value of
        isnsAddressNotification is the zero-length string."
    ::= { isnsNotificationsInfo 2 }

isnsAddressNotification    OBJECT-TYPE
    SYNTAX                 InetAddress
    MAX-ACCESS             accessible-for-notify
    STATUS                 current
    DESCRIPTION
        "Identifies the IP address of the iSNS Server. The format of

```



this object is specified by isnsAddressNotificationType.  
The IP address will always be specified in the notification  
unless an error causes the IP address to not be known."

::= { isnsNotificationsInfo 3 }

```
isnsTcpPortNotification      OBJECT-TYPE
    SYNTAX                    InetPortNumber
    MAX-ACCESS                accessible-for-notify
    STATUS                    current
    DESCRIPTION
```

"Indicates the TCP port the iSNS Server is using,  
or 0 if TCP-based registrations are not supported."

::= { isnsNotificationsInfo 4 }

```
isnsUdpPortNotification      OBJECT-TYPE
    SYNTAX                    InetPortNumber
    MAX-ACCESS                accessible-for-notify
    STATUS                    current
    DESCRIPTION
```

"Indicates the UDP port the iSNS Server is using,  
or 0 if UDP-based registrations are not supported."

::= { isnsNotificationsInfo 5 }

```
--
-- iSNS Notification Block -----
--
```

```
isnsServerStart              NOTIFICATION-TYPE
    OBJECTS {
        isnsInstanceInfo,
        isnsAddressNotificationType,
        isnsAddressNotification,
        isnsTcpPortNotification,
        isnsUdpPortNotification
    }
    STATUS                    current
    DESCRIPTION
```

"This notification is sent when an iSNS Server begins  
operation. The notification provides the following:  
isnsInstanceInfo : iSNS Server textual information  
isnsAddressTypeNotification : iSNS Server address type  
isnsAddressNotification : iSNS Server address  
isnsTcpPortNotification : iSNS Server TCP Port  
isnsUdpPortNotification : iSNS Server UDP Port

"  
::= { isnsNotifications 1 }

```
isnsServerShutdown          NOTIFICATION-TYPE
```

```

OBJECTS {
    isnsInstanceInfo,
    isnsAddressNotificationType,
    isnsAddressNotification,
    isnsTcpPortNotification,
    isnsUdpPortNotification
}

STATUS current

DESCRIPTION
"This notification is sent when an iSNS Server is
shutdown. The notification provides the following:
    isnsInstanceInfo : iSNS Server textual information
    isnsAddressTypeNotification : iSNS Server address type
    isnsAddressNotification : iSNS Server address
    isnsTcpPortNotification : iSNS Server TCP Port
    isnsUdpPortNotification : iSNS Server UDP Port
"

 ::= { isnsNotifications 2 }

-----
--
-- Compliance Information
--

isnsCompliances OBJECT IDENTIFIER ::= { isnsConformance 1 }

isnsIscsiServerCompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
    "Initial compliance statement for an iSNS Server
    providing support to iSCSI clients."
    MODULE -- this module
    MANDATORY-GROUPS {
        isnsServerAttributesGroup,
        isnsServerIscsiControlNodeGroup,
        isnsServerIscsiDdsDdObjGroup,
        isnsServerRegIscsiObjGroup,
        isnsServerNumObjectsGroup,
        isnsNotificationsObjGroup,
        isnsServerNotificationGroup
    }
    OBJECT isnsServerDiscoveryMcGroupType
    SYNTAX InetAddressType { unknown(0), ipv4(1), ipv6(2),
                            ipv4z(3), ipv6z(4) }
    DESCRIPTION
    "Only support for unknown, ipv4, ipv6, ipv4z, ipv6z
    is required."

```

OBJECT isnsServerDiscoveryMcGroupAddress  
SYNTAX InetAddress (SIZE (0 | 4 | 8 | 16 | 20 ))  
DESCRIPTION

"Only addresses for unknown, ipv4, ipv6, ipv4z, ipv6z  
and their related SIZE need to be supported."

OBJECT isnsDdPortalMemberAddressType  
SYNTAX InetAddressType { unknown(0), ipv4(1), ipv6(2),  
ipv4z(3), ipv6z(4) }

DESCRIPTION

"Only support for unknown, ipv4, ipv6, ipv4z, ipv6z  
is required."

OBJECT isnsDdPortalMemberAddress  
SYNTAX InetAddress (SIZE (0 | 4 | 8 | 16 | 20 ))  
DESCRIPTION

"Only addresses for unknown, ipv4, ipv6, ipv4z, ipv6z  
and their related SIZE need to be supported."

OBJECT isnsRegEntityManagementAddressType  
SYNTAX InetAddressType { unknown(0), ipv4(1), ipv6(2),  
ipv4z(3), ipv6z(4) }

DESCRIPTION

"Only support for unknown, ipv4, ipv6, ipv4z, ipv6z  
is required."

OBJECT isnsRegEntityManagementAddress  
SYNTAX InetAddress (SIZE (0 | 4 | 8 | 16 | 20 ))  
DESCRIPTION

"Only addresses for unknown, ipv4, ipv6, ipv4z, ipv6z  
and their related SIZE need to be supported."

OBJECT isnsRegPortalAddressType  
SYNTAX InetAddressType { unknown(0), ipv4(1), ipv6(2),  
ipv4z(3), ipv6z(4) }

DESCRIPTION

"Only support for unknown, ipv4, ipv6, ipv4z, ipv6z  
is required."

OBJECT isnsRegPortalAddress  
SYNTAX InetAddress (SIZE (0 | 4 | 8 | 16 | 20 ))  
DESCRIPTION

"Only addresses for unknown, ipv4, ipv6, ipv4z, ipv6z  
and their related SIZE need to be supported."

OBJECT isnsRegPgPortalAddressType  
SYNTAX InetAddressType { unknown(0), ipv4(1), ipv6(2),  
ipv4z(3), ipv6z(4) }

## DESCRIPTION

"Only support for unknown, ipv4, ipv6, ipv4z, ipv6z is required."

OBJECT isnsRegPgPortalAddress

SYNTAX InetAddress (SIZE (0 | 4 | 8 | 16 | 20 ))

## DESCRIPTION

"Only addresses for unknown, ipv4, ipv6, ipv4z, ipv6z and their related SIZE need to be supported."

OBJECT isnsAddressNotificationType

SYNTAX InetAddressType { unknown(0), ipv4(1), ipv6(2),  
ipv4z(3), ipv6z(4) }

## DESCRIPTION

"Only support for unknown, ipv4, ipv6, ipv4z, ipv6z is required."

OBJECT isnsAddressNotification

SYNTAX InetAddress (SIZE (0 | 4 | 8 | 16 | 20 ))

## DESCRIPTION

"Only addresses for unknown, ipv4, ipv6, ipv4z, ipv6z and their related SIZE need to be supported."

::= { isnsCompliances 1 }

isnsIfcpServerCompliance MODULE-COMPLIANCE

STATUS current

## DESCRIPTION

"Initial compliance statement for an iSNS Server providing support to iFCP Clients."

MODULE -- this module

MANDATORY-GROUPS {  
isnsServerAttributesGroup,  
isnsServerIfcpPortControlNodeGroup,  
isnsServerIfcpDdsDdObjGroup,  
isnsServerRegIfcpObjGroup,  
isnsServerNumObjectsGroup,  
isnsNotificationsObjGroup,  
isnsServerNotificationGroup  
}

OBJECT isnsServerDiscoveryMcGroupType

SYNTAX InetAddressType { unknown(0), ipv4(1), ipv6(2),  
ipv4z(3), ipv6z(4) }

## DESCRIPTION

"Only support for unknown, ipv4, ipv6, ipv4z, and ipv6z is required."

OBJECT isnsServerDiscoveryMcGroupAddress

SYNTAX InetAddress (SIZE (0 | 4 | 8 | 16 | 20 ))

## DESCRIPTION

"Only addresses for unknown, ipv4, ipv6, ipv4z, ipv6z, and their related SIZE need to be supported."

OBJECT isnsDdPortalMemberAddressType

SYNTAX InetAddressType { unknown(0), ipv4(1), ipv6(2),  
ipv4z(3), ipv6z(4) }

## DESCRIPTION

"Only support for unknown, ipv4, ipv6, ipv4z, and ipv6z is required."

OBJECT isnsDdPortalMemberAddress

SYNTAX InetAddress (SIZE (0 | 4 | 8 | 16 | 20 ))

## DESCRIPTION

"Only addresses for unknown, ipv4, ipv6, ipv4z, ipv6z, and their related SIZE need to be supported."

OBJECT isnsRegEntityManagementAddressType

SYNTAX InetAddressType { unknown(0), ipv4(1), ipv6(2),  
ipv4z(3), ipv6z(4) }

## DESCRIPTION

"Only support for unknown, ipv4, ipv6, ipv4z, and ipv6z is required."

OBJECT isnsRegEntityManagementAddress

SYNTAX InetAddress (SIZE (0 | 4 | 8 | 16 | 20 ))

## DESCRIPTION

"Only addresses for unknown, ipv4, ipv6, ipv4z, ipv6z, and their related SIZE need to be supported."

OBJECT isnsRegPortalAddressType

SYNTAX InetAddressType { unknown(0), ipv4(1), ipv6(2),  
ipv4z(3), ipv6z(4) }

## DESCRIPTION

"Only support for unknown, ipv4, ipv6, ipv4z, and ipv6z is required."

OBJECT isnsRegPortalAddress

SYNTAX InetAddress (SIZE (0 | 4 | 8 | 16 | 20 ))

## DESCRIPTION

"Only addresses for unknown, ipv4, ipv6, ipv4z, ipv6z, and their related SIZE need to be supported."

OBJECT isnsRegFcNodeAddressType

SYNTAX InetAddressType { unknown(0), ipv4(1), ipv6(2),  
ipv4z(3), ipv6z(4) }

## DESCRIPTION

"Only support for unknown, ipv4, ipv6, ipv4z, and ipv6z

is required."

OBJECT isnsRegFcNodeAddress

SYNTAX InetAddress (SIZE (0 | 4 | 8 | 16 | 20 ))

DESCRIPTION

"Only addresses for unknown, ipv4, ipv6, ipv4z, ipv6z, and their related SIZE need to be supported."

OBJECT isnsRegFcPortAddressType

SYNTAX InetAddressType { unknown(0), ipv4(1), ipv6(2),  
ipv4z(3), ipv6z(4) }

DESCRIPTION

"Only support for unknown, ipv4, ipv6, ipv4z, and ipv6z is required."

OBJECT isnsRegFcPortAddress

SYNTAX InetAddress (SIZE (0 | 4 | 8 | 16 | 20 ))

DESCRIPTION

"Only addresses for unknown, ipv4, ipv6, ipv4z, ipv6z, and their related SIZE need to be supported."

OBJECT isnsAddressNotificationType

SYNTAX InetAddressType { unknown(0), ipv4(1), ipv6(2),  
ipv4z(3), ipv6z(4) }

DESCRIPTION

"Only support for unknown, ipv4, ipv6, ipv4z, and ipv6z is required."

OBJECT isnsAddressNotification

SYNTAX InetAddress (SIZE (0 | 4 | 8 | 16 | 20 ))

DESCRIPTION

"Only addresses for unknown, ipv4, ipv6, ipv4z, ipv6z, and their related SIZE need to be supported."

::= { isnsCompliances 2 }

isnsGroups OBJECT IDENTIFIER ::= { isnsConformance 2 }

isnsServerAttributesGroup OBJECT-GROUP

OBJECTS {

isnsServerName,  
isnsServerIsnsVersion,  
isnsServerVendorInfo,  
isnsServerPhysicalIndex,  
isnsServerTcpPort,  
isnsServerUdpPort,  
isnsServerDiscontinuityTime,  
isnsServerRole,  
isnsServerDiscoveryMethodsEnabled,

```

        isnsServerDiscoveryMcGroupType,
        isnsServerDiscoveryMcGroupAddress,
        isnsServerEsiNonResponseThreshold,
        isnsServerEnableControlNodeMgtScn,
        isnsServerDefaultDdDdsStatus,
        isnsServerUpdatedDdDdsSupported,
        isnsServerUpdatedDdDdsEnabled
    }
    STATUS current
    DESCRIPTION
    "iSNS Server attributes."
    ::= { isnsGroups 1 }

isnsServerNumObjectsGroup      OBJECT-GROUP
    OBJECTS {
        isnsNumDds,
        isnsNumDd,
        isnsNumEntities,
        isnsNumPortals,
        isnsNumPortalGroups,
        isnsNumIscsiNodes,
        isnsNumFcPorts,
        isnsNumFcNodes,
        isnsRegEntityInfoNumPortals,
        isnsRegEntityInfoNumPortalGroups,
        isnsRegEntityInfoNumIscsiNodes,
        isnsRegEntityInfoNumFcPorts,
        isnsRegEntityInfoNumFcNodes
    }
    STATUS current
    DESCRIPTION
    "Managed objects indicating the number of registered objects
    in an iSNS Server or the number of registered objects
    associated with a registered Entity.  These managed objects
    are optional to implement."
    ::= { isnsGroups 2 }

isnsServerIscsiControlNodeGroup  OBJECT-GROUP
    OBJECTS {
        isnsControlNodeIscsiNodeName,
        isnsControlNodeIscsiIsRegistered,
        isnsControlNodeIscsiRcvMgtSCN
    }
    STATUS current
    DESCRIPTION
    "iSNS Server iSCSI control node managed objects."
    ::= { isnsGroups 3 }

```

```

isnsServerIfcpPortControlNodeGroup    OBJECT-GROUP
    OBJECTS {
        isnsControlNodeFcPortIsRegistered,
        isnsControlNodeFcPortRcvMgtSCN
    }
    STATUS current
    DESCRIPTION
    "iSNS Server iFCP Port control node managed objects."
    ::= { isnsGroups 4 }

isnsServerIscliDdsDdObjGroup    OBJECT-GROUP
    OBJECTS {
        isnsDdsSymbolicName,
        isnsDdsStatus,
        isnsDdsMemberSymbolicName,
        isnsDdSymbolicName,
        isnsDdFeatures,
        isnsDdIscliMemberName,
        isnsDdIscliMemberIsRegistered,
        isnsDdPortalMemberAddressType,
        isnsDdPortalMemberAddress,
        isnsDdPortalMemberPortType,
        isnsDdPortalMemberPort,
        isnsDdPortalMemberIsRegistered
    }
    STATUS current
    DESCRIPTION
    "iSNS Server DDS and DD managed objects for iSCSI."
    ::= { isnsGroups 5 }

isnsServerIfcpDdsDdObjGroup    OBJECT-GROUP
    OBJECTS {
        isnsDdsSymbolicName,
        isnsDdsStatus,
        isnsDdSymbolicName,
        isnsDdFeatures,
        isnsDdPortalMemberAddressType,
        isnsDdPortalMemberAddress,
        isnsDdPortalMemberPortType,
        isnsDdPortalMemberPort,
        isnsDdPortalMemberIsRegistered,
        isnsDdFcPortMemberIsRegistered
    }
    STATUS current
    DESCRIPTION
    "iSNS Server DDS and DD managed objects for iFCP."
    ::= { isnsGroups 6 }

```



```

isnsServerRegIscsiObjGroup    OBJECT-GROUP
    OBJECTS {
        isnsRegEntityEID,
        isnsRegEntityProtocol,
        isnsRegEntityManagementAddressType,
        isnsRegEntityManagementAddress,
        isnsRegEntityTimestamp,
        isnsRegEntityVersionMin,
        isnsRegEntityVersionMax,
        isnsRegEntityRegistrationPeriod,
        isnsRegEntityInfoNumPortals,
        isnsRegEntityInfoNumPortalGroups,
        isnsRegEntityInfoNumIscsiNodes,
        isnsRegEntityInfoNumFcPorts,
        isnsRegEntityInfoNumFcNodes,
        isnsRegPortalAddressType,
        isnsRegPortalAddress,
        isnsRegPortalPortType,
        isnsRegPortalPort,
        isnsRegPortalSymbolicName,
        isnsRegPortalEsiInterval,
        isnsRegPortalEsiPortType,
        isnsRegPortalEsiPort,
        isnsRegPortalScnPortType,
        isnsRegPortalScnPort,
        isnsRegPortalSecurityInfo,
        isnsRegPgIscsiNodeIndex,
        isnsRegPgIscsiName,
        isnsRegPgPortalPortalIndex,
        isnsRegPgPortalAddressType,
        isnsRegPgPortalAddress,
        isnsRegPgPortalPortType,
        isnsRegPgPortalPort,
        isnsRegPgPGT,
        isnsRegIscsiNodeName,
        isnsRegIscsiNodeType,
        isnsRegIscsiNodeAlias,
        isnsRegIscsiNodeScnTypes,
        isnsRegIscsiNodeWwnToken,
        isnsRegIscsiNodeAuthMethod
    }

    STATUS                current

    DESCRIPTION
    "iSNS Server registered iSCSI managed objects."
        ::= { isnsGroups 7 }

isnsServerRegIfcpObjGroup    OBJECT-GROUP
    OBJECTS {

```

```
isnsRegEntityEID,  
isnsRegEntityProtocol,  
isnsRegEntityManagementAddressType,  
isnsRegEntityManagementAddress,  
isnsRegEntityTimestamp,  
isnsRegEntityVersionMin,  
isnsRegEntityVersionMax,  
isnsRegEntityRegistrationPeriod,  
isnsRegEntityInfoNumPortals,  
isnsRegEntityInfoNumPortalGroups,  
isnsRegEntityInfoNumIscsiNodes,  
isnsRegEntityInfoNumFcPorts,  
isnsRegEntityInfoNumFcNodes,  
isnsRegPortalAddressType,  
isnsRegPortalAddress,  
isnsRegPortalPortType,  
isnsRegPortalPort,  
isnsRegPortalSymbolicName,  
isnsRegPortalEsiInterval,  
isnsRegPortalEsiPortType,  
isnsRegPortalEsiPort,  
isnsRegPortalScnPortType,  
isnsRegPortalScnPort,  
isnsRegPortalSecurityInfo,  
isnsRegFcPortID,  
isnsRegFcPortType,  
isnsRegFcPortSymbolicName,  
isnsRegFcPortFabricPortWwn,  
isnsRegFcPortHA,  
isnsRegFcPortAddressType,  
isnsRegFcPortAddress,  
isnsRegFcPortFcCos,  
isnsRegFcPortFc4Types,  
isnsRegFcPortFc4Descr,  
isnsRegFcPortFc4Features,  
isnsRegFcPortScnTypes,  
isnsRegFcPortRole,  
isnsRegFcPortFcNodeWwnn,  
isnsRegFcPortPpnWwn,  
isnsRegFcNodeSymbolicName,  
isnsRegFcNodeAddressType,  
isnsRegFcNodeAddress,  
isnsRegFcNodeIPA,  
isnsRegFcNodeProxyIscsiName,  
isnsRegFcNodeNumFcPorts,  
isnsRegFcNodePortEntityIndex  
}  
STATUS current
```

```

        DESCRIPTION
        "iSNS Server registered iFCP managed objects."
        ::= { isnsGroups 8 }

isnsNotificationsObjGroup      OBJECT-GROUP
    OBJECTS {
        isnsInstanceInfo,
        isnsAddressNotificationType,
        isnsAddressNotification,
        isnsTcpPortNotification,
        isnsUdpPortNotification
    }
    STATUS current
    DESCRIPTION
    "iSNS Notification managed objects."
    ::= { isnsGroups 9 }

isnsServerNotificationGroup    NOTIFICATION-GROUP
    NOTIFICATIONS {
        isnsServerStart,
        isnsServerShutdown
    }
    STATUS current
    DESCRIPTION
    "iSNS Server Notification managed objects."
    ::= { isnsGroups 10 }
END

```

## 6. IANA Considerations

The MIB module in this document uses the following IANA-assigned OBJECT IDENTIFIER values recorded in the SMI Numbers registry:

Descriptor -----	OBJECT IDENTIFIER value -----
isnsMIB	{ mib-2 163 }

This RFC utilizes the IANA registry of iSNS parameters. This registry was created for the iSNS Specification [RFC4171], and is located at <http://www.iana.org/assignments/isns-parameters>. Specifically, the isnsRegEntityProtocol values used in the MIB module are the values for the Block Storage Protocols that IANA assigns and documents in <http://www.iana.org/assignments/isns-parameters>.

## 7. Security Considerations

There are no management objects defined in this MIB module that have a MAX-ACCESS clause of read-write and/or read-create. So, if this MIB module is implemented correctly, then there is no risk that an intruder can alter or create any management objects of this MIB module via direct SNMP SET operations.

Some of the readable objects in this MIB module (i.e., objects with a MAX-ACCESS other than not-accessible) may be considered sensitive or vulnerable in some network environments. It is thus important to control even GET and/or NOTIFY access to these objects and possibly to even encrypt the values of these objects when sending them over the network via SNMP. These are the tables and objects and their sensitivity/vulnerability:

The `isnsDdsMemberTable` contains information about which Discovery Domains may be enabled at the same time.

The `isnsDdTable` contains information about Discovery Domains, containing storage nodes with an ability to communicate and exchange storage data.

The `isnsDdIscsiMemberTable` indicates which iSCSI nodes are contained in which Discovery Domains.

The `isnsDdPortalMemberTable` indicates which iSCSI portals are contained in which Discovery Domains.

The `isnsDdFcPortMemberTable` indicates which iFCP FC N\_Ports are contained in which Discovery Domains.

The `isnsControlNodeIscsiTable` indicates which iSCSI nodes have the ability to possibly control an iSNS server.

The `isnsControlNodeFcPortTable` indicates which iFCP FC N\_Ports have the ability to possibly control an iSNS server.

The above object tables provide information about storage objects sessions, and can indicate to a user who is communicating and exchanging storage data.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPsec), even then, there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

It is RECOMMENDED that implementers consider the security features as provided by the SNMPv3 framework (see [RFC3410], section 8), including full support for the SNMPv3 cryptographic mechanisms (for authentication and privacy).

Further, deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

## 8. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2578] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case J., Rose, M., and S. Waldbusser, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.
- [RFC2579] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M., and S. Waldbusser, "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.
- [RFC2580] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M., and S. Waldbusser, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.
- [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.
- [RFC3720] Satran, J., Meth, K., Sapuntzakis, C., Chadalapaka, M., and E. Zeidner, "Internet Small Computer Systems Interface (iSCSI)", RFC 3720, March 2004.
- [RFC4001] Daniele, M., Haberman, B., Routhier, S., and J. Schoenwaelder, "Textual Conventions for Internet Network Addresses", RFC 4001, February 2005.
- [RFC4044] McCloghrie, K., "Fibre Channel Management MIB", RFC 4044, May 2005.

- [RFC4133] McCloghrie, K. and A. Bierman, "Entity MIB (Version 3)", RFC 4133, August 2005.
- [RFC4171] Tseng, J., Gibbons, K., Travostino, F., Du Laney, C., and J. Souza, "Internet Storage Name Service (iSNS)", RFC 4171, September 2005.
- [RFC4172] Monia, C., Mullendore, R., Travostino, F., Jeong, W., and M. Edwards, "iFCP - A Protocol for Internet Fibre Channel Storage Networking", RFC 4172, September 2005.

## 9. Informative References

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

## 10. Acknowledgements

This memo is a product of the IP Storage (IPS) working group within the Internet Engineering Task Force.

We wish to acknowledge the contributions and comments from the IPS WG, including the following:

IPS WG Chair: David Black  
Former Editors: Josh Tseng and Tom McSweeney  
MIB Editors: Keith McCloghrie and Bert Wijnen

## Authors' Addresses

Kevin Gibbons  
2Wire, Inc.  
1704 Automation Parkway  
San Jose, CA 95131  
USA  
Tel: +1 408-895-1387  
Fax: +1 408-428-9590  
EMail: kgibbons@yahoo.com

G.D. Ramkumar  
SnapTell, Inc.  
2741 Middlefield Rd, Suite 200  
Palo Alto, CA 94306  
USA  
Tel: +1 650-326-7627  
Fax: +1 650-326-7620  
EMail: gramkumar@stanfordalumni.org

Scott Kipp  
Brocade  
4 McDATA Pkwy  
Broomfield, CO 80021  
USA  
Tel: +1 720-558-3452  
Fax: +1 720-558-8999  
EMail: skipp@brocade.com

## Full Copyright Statement

Copyright (C) The IETF Trust (2007).

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, THE IETF TRUST 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](mailto:ietf-ipr@ietf.org).

## Acknowledgement

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



