

Network Working Group  
Request for Comments: 4455  
Category: Standards Track

M. Hallak-Stamler  
Sanrad Intelligent Storage  
M. Bakke  
Cisco Systems, Inc.  
Y. Lederman  
Siliquent Technologies  
M. Krueger  
Hewlett-Packard  
K. McCloghrie  
Cisco Systems, Inc.  
April 2006

## Definition of Managed Objects for Small Computer System Interface (SCSI) Entities

### Status of This Memo

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

### Copyright Notice

Copyright (C) The Internet Society (2006).

### Abstract

This memo defines a portion of the Management Information Base (MIB), for use with network management protocols in the Internet community. In particular, it describes managed objects for Small Computer System Interface (SCSI) entities, independently of the interconnect subsystem layer.

### Table of Contents

1. The Internet-Standard Management Framework .....	3
2. Requirements Notation .....	3
3. Overview .....	3
3.1. Introduction .....	4
3.2. SCSI Terminology .....	6
3.2.1. SCSI Application Layer .....	6
3.2.2. SCSI Device .....	6
3.2.3. SCSI Port .....	6
3.2.4. SCSI Initiator Device .....	7
3.2.5. SCSI Initiator Port .....	7

3.2.6. SCSI Target Device .....	7
3.2.7. SCSI Target Port .....	7
3.2.8. Logical Units .....	7
3.2.9. Logical Unit Number .....	7
3.2.10. Interconnect Subsystem .....	7
3.2.11. Device Server .....	8
3.2.12. Task Manager .....	8
3.2.13. SCSI Instance .....	8
3.3. SCSI MIB Module Implementation .....	8
3.4. Bridging and Virtualization .....	10
3.5. SCSI Command MIB Module .....	11
4. Structure of the MIB .....	11
4.1. The SCSI Device Group .....	11
4.2. The Initiator Group .....	11
4.3. The Target Group .....	11
4.4. The Discovery Group .....	12
4.5. The LUN Map Group .....	12
4.6. The Target Statistic Group .....	12
4.7. The Target High Speed Statistic Group .....	12
4.8. The LUN Map Statistics Group .....	12
4.9. The LUN Map Statistics High Speed Group .....	13
4.10. The Initiator Statistics Group .....	13
4.11. The Initiator High Speed Statistic Group .....	13
4.12. The Discovery Statistics Group .....	13
4.13. The Discovery Statistics High Speed Group .....	14
4.14. The Device Statistics Group .....	14
5. Relationships in This MIB .....	14
6. Relationship to Other MIBs .....	16
6.1. Host Resource MIB .....	16
6.2. iSCSI MIB Module .....	16
7. Miscellaneous Details .....	16
7.1. Names and Identifiers .....	16
7.2. Logical Unit Number .....	16
7.3. Notifications .....	16
7.4. SCSI Domains .....	17
7.5. Counters: 32 Bits and 64 Bits .....	17
7.6. Local versus Remote Entities .....	18
8. Abbreviations .....	18
9. Object Definitions .....	18
10. Object Population Example: SCSI Target and Initiator	
Devices on a pSCSI Bus .....	76
10.1. scsiInstance Table: .....	77
10.2. scsiDevice Table: .....	77
10.3. scsiPort Table: .....	77
10.4. scsiTransport Table: .....	77
10.5. scsiIntrDev Table: .....	78
10.6. scsiInitiatorPort Table: .....	78
10.7. scsiDscTgt Table: .....	78

10.8. scsiDscLUN:	78
10.9. scsiDscLUNIdentifier:	79
10.10. scsiAttTgtPort Table:	79
10.11. scsiTgtDev Table:	79
10.12. scsiTgtPort Table:	80
10.13. scsiLU Table:	80
10.14. scsiLuId Table:	80
10.15. scsiLunMap Table:	81
10.16. scsiAuthorizedIntr Table:	81
10.17. scsiAttIntrPort Table:	81
11. Security Considerations	81
12. Acknowledgements	84
13. IANA Considerations	84
14. References	84
14.1. Normative References	84
14.2. Informative References	85

## 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. Requirements Notation

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. Overview

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes a set of managed objects to configure and monitor Small Computer System Interface entities (SCSI entities), i.e., SCSI target devices and SCSI initiator devices and SCSI ports.

SCSI is a client-server protocol in which application clients within a SCSI initiator device (client) issue service requests to logical units contained in a SCSI target device(server).

This MIB module is based on documents defined by the ANSI T10 Technical Committee, specifically the SCSI Architecture Model - 2 [SAM2] and SCSI Primary Commands - 2 [SPC2].

The [SAM2] standard is the primary source for the SCSI architecture discussion in this document and the terminology used in this MIB module.

### 3.1. Introduction

In the late 1970s, a firm called Shugart Associates started to have some considerable success with a peripheral interface definition in what became the PC marketplace, and this interface was adopted and extended by an open standards committee to form the Small Computer Systems Interface (SCSI). SCSI defines an 8-bit-wide multi-drop "bus" structure, which could interconnect a total of eight peripherals and computer systems.

It is important to realize that initially SCSI standardized only the "physical connection", i.e., the connectors, cables, and interface signals. Thus, even though a peripheral could be connected to multiple systems, the information that flowed across the interface was different in each case. This was addressed some five years later by the definition of a Common Command Set, and with this definition in place it was possible for the first time to develop a peripheral with both a common interface and common operating firmware for connection to multiple systems.

The physical interface of SCSI continued to be developed throughout the 1980s with the addition of fast (up to 10 megabytes/s) and wide (16 bits) variants, but the distance supported remained a maximum of 25 meters (from one end of the bus to another), and indeed some of the faster variants supported much less than that distance. The command set development continued, with special commands for tapes, printers, and even processors being added to the original disk-oriented set. So successful was SCSI in the 1980s that the majority of the available Operating Systems incorporated support for the SCSI command set as standard.

However, at the end of the 1980s the distance, speed, and number of devices supported by SCSI were starting to become significant impediments to systems design, and although the "information explosion" had not yet started in earnest, it was already being anticipated. At the same time, the serial interface technologies

developed for Local Area Networks such as Ethernet, and the fibre optics technologies that were first deployed in telecommunications applications were starting to appear sufficiently rugged and low cost for use in peripheral interface applications. Thus, a standards project was begun in 1988 to develop a new serial, fibre-optic interface to carry the SCSI command sets and other peripheral protocols. This interface eventually became known as Fibre Channel (FC), and it is based on an architecture centered around an abstractly defined "fabric", which may be a switch or a loop connection. MIB modules for various FC equipments are already in existence.

In order to support the new interfaces, it was necessary to completely reorganize the SCSI standards and definitions. The command sets were separated from the physical interface definitions, and a SCSI Architectural Model (SAM) was created to define the interaction between the various standards. It is a key to understanding SAM to realize that it was first created approximately 10 years AFTER the first SCSI products were shipped!

The most recent development in this saga occurred in 2000 when an IETF Working Group was formed to address, among other things, a definition for transporting the SCSI command sets directly over a TCP/IP infrastructure. This effort is known as iSCSI [[RFC3720](#)], and an iSCSI MIB module is already under development [[ISCSI](#)].

Most of the projects are in T10, except Fibre Channel, which is defined by T11 and IEEE defines 1394.

The SCSI MIB module represents the SCSI protocol layer common to all SCSI command sets and transports. It does not represent the command sets and transports themselves. These should appear in other MIB modules specific to the transport or command set. The following illustration shows the relationships between the various actual and possible SCSI-related MIB modules.

SCSI Command Sets	Higher-level MIBs, specific to command sets, disk, tape, etc.			
SCSI	SCSI MIB			
SCSI Transport Protocols	iSCSI MIB	FCP MIB	SPI MIB	Other MIBs
SCSI Interconnect	TCP MIB	Fibre Channel MIBs	Other Interconnect MIBs	

An iSCSI MIB module [[ISCSI](#)] and a Fibre Channel interconnect MIB module [[RFC4044](#)] are currently being developed. No development is currently planned for standard command-set-specific or device-specific MIBs.

The TCP-MIB [[RFC4022](#)] is already a proposed standard [RFC 4022](#).

### 3.2. SCSI Terminology

The following sections explain some of the SCSI terminology, which is used later in defining the MIB module. For the authoritative definitions of these terms, see SAM-2 [[SAM2](#)].

#### 3.2.1. SCSI Application Layer

The protocols and procedures that implement or invoke SCSI commands and task management functions by using services provided by a SCSI transport protocol layer.

#### 3.2.2. SCSI Device

A SCSI device is an entity that contains one or more SCSI ports that are connected to a service delivery subsystem and supports a SCSI application protocol.

#### 3.2.3. SCSI Port

A SCSI port is a device-resident entity that connects the application client, device server, or task manager to the service delivery subsystem through which requests and responses are routed. A SCSI port is synonymous with port and either a SCSI initiator port or a SCSI target port.

#### 3.2.4. SCSI Initiator Device

A SCSI initiator device contains application clients and SCSI initiator ports that originate device service and task management requests to be processed by a SCSI target device. When used, this term refers to SCSI initiator devices or SCSI target/initiator devices that are using the SCSI target/initiator port as a SCSI initiator port.

#### 3.2.5. SCSI Initiator Port

A SCSI initiator port acts as the connection between application clients and the service delivery subsystem through which requests and responses are routed. In all cases when this term is used, it refers to an initiator port or a SCSI target/initiator port operating as a SCSI initiator port.

#### 3.2.6. SCSI Target Device

A SCSI target device contains logical units and SCSI target ports that receive device service and task management requests for processing. When used, this term refers to SCSI target devices or SCSI target/initiator devices that are using the SCSI target/initiator port as a SCSI target port.

#### 3.2.7. SCSI Target Port

A SCSI target port contains a task router and acts as the connection between device servers and task managers and the service delivery subsystem through which requests and responses are routed. When this term is used, it refers to a SCSI target port or a SCSI target/initiator port operating as a SCSI target port.

#### 3.2.8. Logical Units

A logical unit is an entity residing in the SCSI target device that implements a device model and processes SCSI commands sent by an application client.

#### 3.2.9. Logical Unit Number

A Logical Unit Number or LUN is a 64-bit identifier for a logical unit.

#### 3.2.10. Interconnect Subsystem

An interconnect subsystem is one or more interconnects that appear as a single path for the transfer of information between SCSI devices.

### 3.2.11. Device Server

A device server is an object within the logical unit that processes SCSI tasks according to the rules for task management.

### 3.2.12. Task Manager

A task manager is a server within the SCSI target device that processes task management functions.

### 3.2.13. SCSI Instance

A "SCSI instance" is a distinct SCSI entity within a managed system. Whereas most implementations will have just one SCSI instance, the MIB module allows for multiple (virtual) instances, such that a large system can be "partitioned" into multiple, distinct virtual systems.

For example, in a host, it allows multiple vendors' implementations of the MIB module to co-exist under a single SNMP agent through each vendor's implementation being a different SCSI instance. It also allows a single SNMP agent to represent multiple subsystems each of which has its own SCSI instance.

## 3.3. SCSI MIB Module Implementation

The SCSI MIB module is a basic building block to use in the various SCSI management scenarios. This module is intended to be implemented in every SCSI entity in a managed system. A SCSI entity can be a SCSI initiator device, SCSI target device or SCSI initiator and Target device. Since SCSI (storage) networking devices may contain more than one SCSI entity, it is possible that more than one SCSI instance will reside in a single device.

In small-scale environments, a single network management station (NMS) may have SNMP access to both SCSI initiator devices and SCSI target devices. However, if the SCSI target devices, or virtualized target devices, are being provided as a service, it is more likely that the provider of the service owns and manages the SCSI target devices and that the consumer of the service owns and manages the SCSI initiator devices. In this case, the service provider NMS and the consumer NMS may have only allowed SNMP access to the SCSI target devices and the SCSI initiator devices, respectively.

The figures in this chapter describe the location of the SCSI MIB module implementations in the various SCSI management scenarios. The locations of the SCSI SNMP agent implementing the SCSI MIB module are denoted with '\*'.



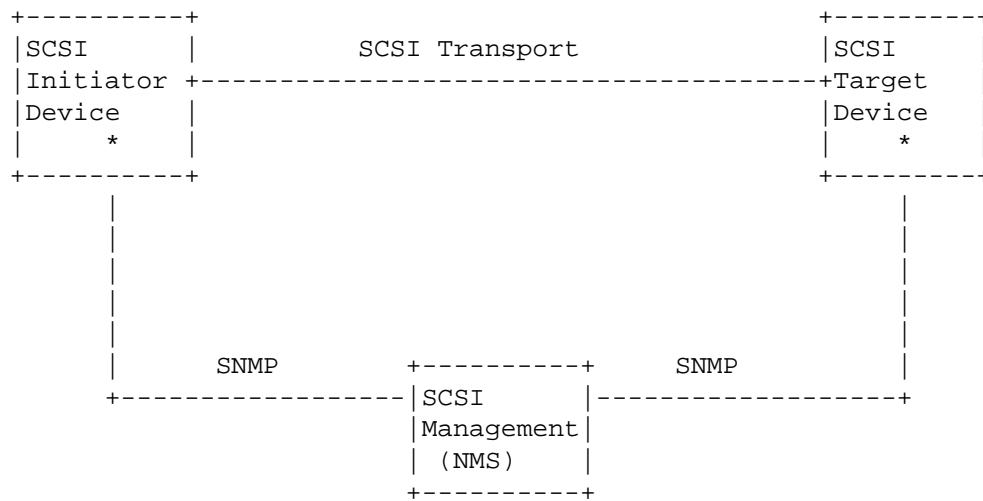


Figure 1. Single SCSI Initiator Device and Single SCSI Target Device

Figure 1 describes a simple SCSI management scenario of a SCSI initiator device, a SCSI target device, and a management station. In this scenario, there are two SNMP agents, each containing its SCSI instance and its respective objects. As the SCSI target device and SCSI initiator device are interconnected, their target and initiator port objects will be complementary.

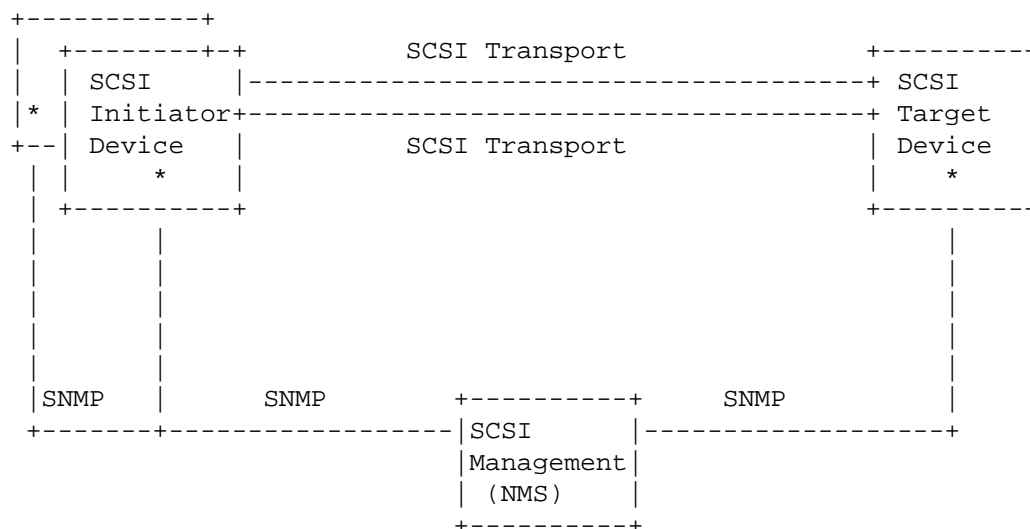


Figure 2. Multiple Hosts and a Single Target Device

Figure 2 adds another SCSI initiator device, to the SCSI network, which connects to the same SCSI target device. The additional SCSI initiator device also has an SNMP agent implementing the SCSI MIB module. In this case, the SCSI target device's MIB module will show that two SCSI initiator devices are attached to it.

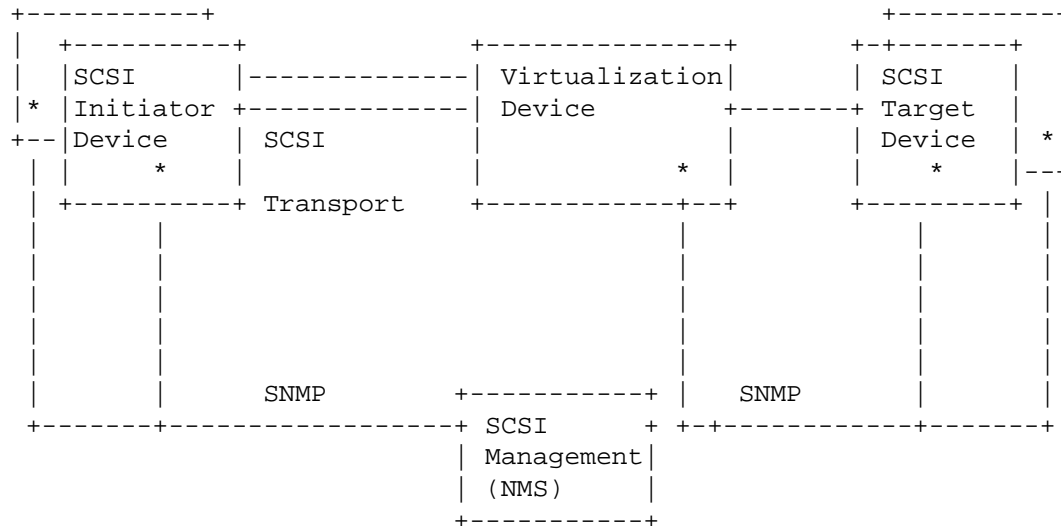


Figure 3. Multiple Hosts, Virtualization Device and Multiple SCSI Target Devices

Figure 3 adds an in-band virtualization device that encapsulates, and possibly modifies, the SCSI target devices' representation to the SCSI Initiator devices. It is common practice for an in-band virtualization device to include both SCSI target and initiator device functionality. Therefore, its SCSI MIB module implementation includes both the SCSI Target device and Initiator device objects. It should be noted that the Virtualization device might implement additional proprietary MIB modules, as the SCSI MIB module does not distinguish between physical and virtual SCSI entities.

### 3.4. Bridging and Virtualization

Storage virtualization is a concept that abstracts storage resources in such a way that, storage entities are provided as pool of logical entities.

Usually, the virtualization process is transparent to the storage users (i.e., hosts). Virtualization normally affects the SCSI entities represented to SCSI initiator devices. However, the SCSI MIB module enables the representation of SCSI entities and their respective status, including error and performance-monitoring

statistics. It should be possible to perform a limited number of configuration modification and diagnostic actions.

The SCSI entities embodied in the bridging and virtualization devices can be represented by the SCSI MIB module. However, the configuration of bridging and virtualization devices is beyond the above-described scope and therefore should be provided through other MIB modules.

### 3.5. SCSI Command MIB Module

The management of SCSI commands is beyond the scope of this MIB module. Future SCSI Command MIB module can link to this MIB module, through the use of Object Identifiers (OIDs) or INDEX values of appropriate tables.

## 4. Structure of the MIB

This MIB module contains fourteen conformance groups:

### 4.1. The SCSI Device Group

The `scsiDeviceGroup` group contains the objects general to each SCSI instance: instance, device, and port objects. It contains also the objects referring to the transport(s) used by those SCSI instances. This group is mandatory for all SCSI managed system.

Alias objects are provided for SCSI instances and SCSI devices to enable administrators to identify them. These objects contain human-readable administrative text strings, and hence use the `SnmpAdminString` textual convention from [RFC3411].

### 4.2. The Initiator Group

The `scsiInitiatorDeviceGroup` contains all the managed information related to a local SCSI initiator device and port. In addition, it contains the managed objects referring to the monitored attached SCSI target devices. Any managed system acting as a SCSI initiator or target/initiator device and port MUST support this group.

### 4.3. The Target Group

The `scsiTargetDeviceGroup` contains all the managed objects related to a local SCSI target device, a local SCSI target port, monitored attached initiator ports, logical units, and logical unit identifiers.

Managed systems acting as a SCSI target or target/initiator device and port must support this group.

#### 4.4. The Discovery Group

The `scsiDiscoveryGroup` group is a collection of managed objects referring to remote SCSI target devices, remote SCSI target ports, remote logical units, and remote logical unit identifiers discovered by or configured to a managed system acting as a SCSI initiator device.

Managed systems acting as a SCSI initiator device and port and supporting remote SCSI target devices or ports configuration or discovery should implement this group.

#### 4.5. The LUN Map Group

The `scsiLunMapGroup` group is a collection of managed objects allowing mapping between SCSI target devices, logical units, and logical unit numbers in one side to remote authorized SCSI initiator devices or ports in another side.

Managed systems supporting this mapping should implement the `scsiLunMapGroup`.

#### 4.6. The Target Statistic Group

The `scsiTargetDevStatsGroup` group is a collection of managed objects representing various statistics referring to a SCSI target device or port. Managed systems acting as a SCSI target device and port supporting statistics should implement this group.

#### 4.7. The Target High Speed Statistic Group

The `scsiTargetDevHSStatsGroup` group is a collection of managed objects representing various statistics referring to a SCSI target device or port. It provides support for systems that can quickly generate countable information because they run at high speed.

Managed systems acting as a SCSI target device and port and running at high speed supporting should implement this group.

#### 4.8. The LUN Map Statistics Group

The `scsiLunMapStatsGroup` group is a collection of managed objects representing various statistics referring to remote authorized SCSI initiator devices or ports.

Managed systems acting as a SCSI target device and port and able to gather statistics on remote SCSI initiator devices or ports should implement this group.

#### 4.9. The LUN Map Statistics High Speed Group

The `scsiLunMapHSStatsGroup` group is a collection of managed objects representing various statistics referring to remote authorized SCSI initiator devices or ports. It provides support for systems that can quickly generate countable information because they run at high speed.

Managed systems acting as a SCSI target device and port and able to gather statistics on remote SCSI initiator devices or ports and running at high speed should implement this group.

#### 4.10. The Initiator Statistics Group

The `scsiInitiatorDevStatsGroup` group is a collection of managed objects representing various statistics referring to a SCSI initiator device or port.

Managed systems acting as a SCSI initiator device and port supporting statistics should implement this group.

#### 4.11. The Initiator High Speed Statistic Group

The `scsiInitiatorDevHSStatsGroup` group is a collection of managed objects representing various statistics referring to a SCSI initiator device or port. It provides support for systems that can quickly generate countable information because they run at high speed.

Managed systems acting as a SCSI initiator device and port and running at high speed supporting should implement this group.

#### 4.12. The Discovery Statistics Group

The `scsiDiscoveryStatsGroup` group is a collection of managed objects representing various statistics referring to remote discovered or configured SCSI target devices or ports.

Managed systems acting as a SCSI initiator device and port and able to gather statistics on remote SCSI target devices or ports should implement this group.

#### 4.13. The Discovery Statistics High Speed Group

The `scsiDiscoveryHSStatsGroup` group is a collection of managed objects representing various statistics referring to remote discovered or configured SCSI target devices or ports. It provides support for systems that can quickly generate countable information because they run at high speed.

Managed systems acting as a SCSI initiator device and port and able to gather statistics on remote SCSI target devices or ports and running at high speed should implement this group.

#### 4.14. The Device Statistics Group

The `scsiDeviceStatGroup` group is a collection of managed objects representing various statistics referring to a SCSI device.

Managed systems able to gather device statistics should implement this group.

### 5. Relationships in This MIB

This section outlines the functionality and the dependency between the MIB tables providing the required management functionality for SCSI initiator and target devices. For specific usage of these tables, the reader should refer to the description of the tables and their respective table entries and attributes.

Following is a list of required SCSI initiator-related features, and the respective tables facilitating this functionality:

- o List all the SCSI initiator ports that should be managed through this MIB module. The table `scsiIntrPortTable` maintains all the SCSI initiator ports for the SCSI initiator devices in the MIB module.
- o Provide a list of all SCSI target ports or SCSI target devices to which a SCSI initiator port can attach. This should prevent a SCSI initiator device or port from attaching to SCSI target devices that should be either invisible or inaccessible to it. The entries in this list can be created either manually or by automatic discovery mechanisms (e.g., SLP, iSNS). The `ScsiDscTgtTable` provides this information. The entries in this table point to the SCSI initiator port, and indicate that the SCSI initiator port can only attach to SCSI target ports or SCSI target devices provided in the respective entries of the `ScsiDscTgtTable`.

This MIB module permits, but does not require, this table to be written via SNMP. There are significant security considerations in allowing writes to this table; see [Section 11](#).

- o The information, for the aforementioned SCSI target ports or SCSI target devices, about the LUs and their respective LUN Ids should be provided. The `scsiDscLunTable` and `scsiDscLunIdTable` maintain this information.
- o The `scsiAttTgtPortTable` provides the information about the SCSI target ports each SCSI initiator port is currently communicating with. This table should be dynamically updated to reflect those connections.

Following is a list of required SCSI target device-related features, and the respective tables facilitating this functionality:

- o List all the SCSI target ports that should be managed through this MIB module. The table `scsiTgtPortTable` maintains all the SCSI target ports for the SCSI target devices in the MIB module.
- o Provide a list of valid SCSI initiator ports or SCSI initiator devices authorized to attach to a SCSI target port. This list should feature the concept of "access lists", which are common in IP routers and switches. The `ScsiAuthorizedIntr` table provides this information. This MIB module permits, but does not require this table to be written via SNMP. There are significant security considerations in allowing writes to this table; see [Section 11](#).
- o It should be possible to specify the list of LUNs exposed to each SCSI initiator port or device, when it is attached to the SCSI target device. SCSI target devices must provide a default list of LUNs. This list of LUNs can either be a unique list for each SCSI initiator device or be the default list. For each entry in the `ScsiAuthorizedIntr` table, a pointer, named `scsiAuthIntrLunMapIndex`, indexing the `ScsiLunMapTable` facilitates this feature.
- o Provide means to monitor all the SCSI initiator ports currently attached to this SCSI target port. The `scsiAttIntrPortTable` provides this information. This table should be dynamically updated to reflect those connections.

## 6. Relationship to Other MIBs

### 6.1. Host Resource MIB

The SCSI MIB module extends objects defined in the host resource MIB module to SCSI-specific entities but does not contain information on software modules such as device drivers. If MIB objects are required for installed packages of SCSI software, then the hrSWInstalledGroup of the Host Resources MIB [RFC2790] are the standard MIB objects to use.

### 6.2. iSCSI MIB Module

The SCSI MIB module defines managed objects for the SCSI protocol layer. The SCSI layer can run on top of several transport layers; iSCSI is one of them. The iSCSI-MIB [iSCSI] is the MIB portion defining the managed objects for the transport called iSCSI. In the same way, a fibre channel or parallel SCSI MIB module would define managed objects for a transport called, respectively, fibre channel or parallel SCSI.

The relationship between the SCSI MIB module and any valid transport MIB module is determined via the SCSI port managed table that has an object pointing to the corresponding row, if any, of the relevant table in a transport MIB module.

## 7. Miscellaneous Details

### 7.1. Names and Identifiers

The names and the identifiers of the SCSI devices, ports, and logical units depend on the underlying transport protocols; their format and length vary accordingly. Please refer to SAM-2 [SAM2] for more details.

### 7.2. Logical Unit Number

The Logical Unit Number is a 64-bit integer. This type does not exist in SMI and therefore, this MIB contains a textual convention defining LUN as an OCTET STRING.

### 7.3. Notifications

Separate SNMP notifications may be enabled/disabled to notify of a change in any of the SCSI device status variables. A notification will be generated theoretically for each occurrence (see restriction



below) of the abnormal status (e.g., if the SCSI device's current status is abnormal and another logical unit changes its status from available to abnormal another notification will occur).

To avoid sending an excessive number of notifications due to multiple errors counted, an SNMP agent implementing the SCSI MIB module should not send more than three SCSI notifications in any 10-second period.

The 3-in-10 rule was chosen because one notification every three seconds was deemed often enough, but if and when two or three different notifications happen at the same time, it would not be desirable to suppress them. Three notifications in 10 seconds is a happy medium, where a short burst of notifications is allowed, without inundating the network and/or destination host with a large number of notifications.

The ultimate control on sending of notifications is in command of the notification generator module specified in [\[RFC3413\]](#).

#### 7.4. SCSI Domains

SAM-2 [\[SAM2\]](#) specifies that devices belong to a domain. However, it is not usually possible to determine this from within a system, so domains are not represented within this MIB module.

#### 7.5. Counters: 32 Bits and 64 Bits

Some counters, in (newer) high-performance systems, can increase at a fast enough rate such that their representation as Counter32s can cause them to "wrap" in less than an hour. The SMIV2 provides Counter64 as the syntax for such counters. However, (older) SNMPv1 implementations cannot support Counter64s. Thus, this MIB module defines such counters as both Counter32s and Counter64's.

The counters in this MIB module that count data are defined in terms of megabytes (i.e., as the number of megabytes of data), such that Counter64s are not required.

However, the counters in this MIB module that count commands, when in use at 5 Gbit/second with 512-byte read/write operations, could wrap within an hour. Therefore, each of these counters will be defined as both a Counter32 and a Counter64, with the latter being mandatory, for system speeds of 4 Gbit/second or higher.

A possible (but not required) implementation strategy is to have the value of each Counter32 be the same value as the low-order 32 bits of the corresponding Counter64.

### 7.6. Local versus Remote Entities

This MIB module qualifies often SCSI entities as local or remote. The local entities are the ones for which the agent is reporting. The remote entities are the ones that the local entities are in communication with via the SCSI protocol.

## 8. Abbreviations

This MIB module will use the following abbreviations:

Inst = Instance  
 Dev = SCSI Device  
 Tgt = SCSI Target Device  
 Intr = SCSI Initiator Device  
 Att = Attached  
 Id = Identifier  
 Dsc = Discovered  
 pSCSI = Parallel SCSI

## 9. Object Definitions

SCSI-MIB DEFINITIONS ::= BEGIN

```

IMPORTS
MODULE-IDENTITY, OBJECT-IDENTITY, OBJECT-TYPE,
NOTIFICATION-TYPE, Integer32, Unsigned32, Counter32,
Counter64, Gauge32,
mib-2 FROM SNMPv2-SMI
TEXTUAL-CONVENTION, TimeStamp, TruthValue,
RowStatus, RowPointer, AutonomousType,
StorageType FROM SNMPv2-TC
MODULE-COMPLIANCE, OBJECT-GROUP,
NOTIFICATION-GROUP FROM SNMPv2-CONF
SnmpAdminString FROM SNMP-FRAMEWORK-MIB;

scsiMIB MODULE-IDENTITY
    LAST-UPDATED "200603300000Z" -- 30th March 2006
    ORGANIZATION "IETF"
    CONTACT-INFO "
        Michele Hallak-Stamler

```

Sanrad Intelligent Network  
27 Habarzel Street  
Tel Aviv, Israel  
Phone: +972 3 7674809  
E-mail: michele@sanrad.com

Yaron Lederman  
Siliquent Technologies Ltd.  
21 Etzel Street  
Ramat Gan, Israel  
Phone: +972 54 5308833  
E-mail: yaronled@bezeqint.net

Mark Bakke  
Postal: Cisco Systems, Inc  
7900 International Drive, Suite 400  
Bloomington, MN  
USA 55425  
E-mail: mbakke@cisco.com

Marjorie Krueger  
Postal: Hewlett-Packard  
8000 Foothills Blvd.  
Roseville, CA 95747  
E-mail: marjorie\_krueger@hp.com

Keith McCloghrie  
Cisco Systems, Inc.  
Postal: 170 West Tasman Drive  
San Jose, CA USA 95134  
Phone: +1 408 526-5260  
E-mail: kzm@cisco.com

"

#### DESCRIPTION

"The SCSI MIB Module.  
Copyright (C) The Internet Society (2006). This version of  
this MIB module is part of [RFC 4455](#); see the RFC  
itself for full legal notices."

#### -- Revision History

REVISION "200603300000Z"  
DESCRIPTION " Initial version published as [RFC 4455](#)."  
::= { mib-2 139}

--\*\*\*\*\* Textual Conventions \*\*\*\*\*  
ScsiLUN ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"This textual convention represents a SCSI Logical Unit Number (LUN). The format of a LUN is documented in Tables A.2 and A.3 of SAM-2 [SAM2]."

REFERENCE

"SCSI Architecture Model-2 (SAM-2), ANSI INCITS 366-2003, T10 Project 1157-D, 12 September 2002 - Annex A [SAM2]"  
SYNTAX OCTET STRING (SIZE ( 2 | 8))

ScsiIndexValue ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUS current

DESCRIPTION

"An arbitrary integer value, greater than zero, for use as a unique index value."  
SYNTAX Unsigned32 (1..4294967295)

ScsiPortIndexValueOrZero ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUS current

DESCRIPTION

"This textual convention is an extension of the ScsiIndexValue convention. The latter defines a greater than zero value used to identify an index. This extension permits the additional value of zero and is applicable only to indices of SCSI port. Usage of the 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 index was unknown, or when none or all indices need to be referenced."  
SYNTAX Unsigned32 (0..4294967295)

ScsiIndexValueOrZero ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUS current

DESCRIPTION

"This textual convention is an extension of the ScsiIndexValue convention. The latter defines a greater than zero value used to identify an index. This extension permits the additional value of zero. Usage of the 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 index was unknown, or when none or all indices need to be referenced."  
SYNTAX Unsigned32 (0..4294967295)

ScsiIdentifier ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"This textual convention represents a generic SCSI port identifier.

The format depends on the transport used and is documented in Tables A.2 and A.3 of SAM-2 [SAM2]."

REFERENCE

"SCSI Architecture Model-2 (SAM-2), ANSI INCITS 366-2003, T10 Project 1157-D, 12 September 2002 - Annex A [SAM2]"

SYNTAX OCTET STRING (SIZE (0..262))

ScsiName ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"This textual convention represents the name of a SCSI initiator device, a SCSI target device, a SCSI initiator port or a SCSI target port.

The format depends on the transport used and is documented in Tables A.4 and A.5 of SAM-2 [SAM2].

Every object defined using this syntax must define whether it is

- a) always used for a port,
- b) always used for a device, or
- c) the circumstances under which it is used for a port or device."

REFERENCE

"SCSI Architecture Model-2 (SAM-2), ANSI INCITS 366-2003, T10 Project 1157-D, 12 September 2002 - Annex A [SAM2]"

SYNTAX OCTET STRING (SIZE (0..262))

ScsiLuNameOrZero ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"This textual convention represents either the name of a SCSI logical unit or a zero-length string. Objects defined with this syntax must specify the meaning of the zero-length string.

The format of the name of a LU is defined as:

- a zero-length octet string or
- a string of eight bytes."

REFERENCE

"SCSI Architecture Model-2 (SAM-2), ANSI INCITS 366-2003, T10 Project 1157-D, 12 September 2002 - Annex A [SAM2]"

SYNTAX OCTET STRING (SIZE (0 | 8))

ScsiDeviceOrPort ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"This type specifies whether a particular configuration is applicable to a port or to a device."

SYNTAX INTEGER {  
    device(1),  
    port(2),  
    other(3)  
}

ScsiIdCodeSet ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUS current

DESCRIPTION

"This textual convention specifies the code set for the identifier contained in an Identification Descriptor returned in a logical unit's Device Identification Page, and is formatted as defined in T10 SPC-2 (see REFERENCE) Table 172 - Code Set"

REFERENCE

"ANSI - SCSI Primary Commands - 2 (SPC-2),  
ANSI INCITS 351-2001, 11 July 2001 Chapter 8: [section 8.4.4](#),  
Vital Product Data Parameters [[SPC2](#)]"

SYNTAX Unsigned32 (0..15)

ScsiIdAssociation ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUS current

DESCRIPTION

"This textual convention specifies what the identifier is associated with (e.g., with the addressed physical/logical device or with a particular port) for the identifier contained in an Identification Descriptor returned in a logical unit's Device Identification Page, and is formatted as defined in T10 SPC-2 (see REFERENCE) Table 173 - Association."

REFERENCE

"ANSI - SCSI Primary Commands - 2 (SPC-2),  
ANSI INCITS 351-2001, 11 July 2001 Chapter 8: [section 8.4.4](#),  
Vital Product Data Parameters [[SPC2](#)]"

SYNTAX Unsigned32 (0..3)

ScsiIdType ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUS current

DESCRIPTION

"This textual convention specifies the type for the identifier contained in an Identification Descriptor returned in a

logical unit's Device Identification Page, and is formatted as defined in T10 SPC-2 (see REFERENCE) table 174 - Identifier Type."

## REFERENCE

"ANSI - SCSI Primary Commands - 2 (SPC-2),  
ANSI INCITS 351-2001, 11 July 2001 Chapter 8: [section 8.4.4](#),  
Vital Product Data Parameters [[SPC2](#)]"  
SYNTAX Unsigned32 (0..15)

ScsiIdValue ::= TEXTUAL-CONVENTION

STATUS current

## DESCRIPTION

"This textual convention represents an identifier. The objects of type ScsiIdCodeSet, ScsiIdAssociation, ScsiIdType define together the format.

The format is the same as contained in an Identification Descriptor returned in a logical unit's Device Identification Page, and is formatted as defined in T10 SPC-2 (see REFERENCE)."

## REFERENCE

"ANSI - SCSI Primary Commands - 2 (SPC-2),  
ANSI INCITS 351-2001, 11 July 2001 Chapter 8: [section 8.4.4](#),  
Vital Product Data Parameters [[SPC2](#)]"  
SYNTAX OCTET STRING (SIZE (0..255))

ScsiHrSWInstalledIndexOrZero ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUS current

## DESCRIPTION

"The index value for a software module's row in the Host Resources MIBs hrSWInstalledTable. A zero value indicates that no row in the hrSWInstalledTable is applicable."

## REFERENCE

"hrSWInstalledTable is defined in the Host Resources MIB, [[RFC2790](#)]."  
SYNTAX Integer32 (0..2147483647)

--\*\*\*\*\* Structure of the MIB \*\*\*\*\*

scsiNotifications OBJECT IDENTIFIER ::= { scsiMIB 0 }

scsiAdmin OBJECT IDENTIFIER ::= { scsiMIB 1 }

scsiObjects OBJECT IDENTIFIER ::= { scsiMIB 2 }

scsiConformance OBJECT IDENTIFIER ::= { scsiMIB 3 }

scsiTransportTypes OBJECT IDENTIFIER ::= { scsiAdmin 1 }

scsiGeneral OBJECT IDENTIFIER ::= { scsiObjects 1 }

scsiInitiatorDevice OBJECT IDENTIFIER ::= { scsiObjects 2 }

scsiTargetDevice OBJECT IDENTIFIER ::= { scsiObjects 3 }

```

scsiLogicalUnit      OBJECT IDENTIFIER ::= { scsiObjects 4 }

--***** Transport Types *****
-- The following object identifiers allow determining the different
-- transports (service delivery subsystems) in use under the SCSI
-- layer.

scsiTransportOther   OBJECT-IDENTITY
    STATUS    current
    DESCRIPTION
        "This identity identifies a transport that has no identity; it
        might happen because the transport is unknown or might not
        have been defined when this MIB module was created."
    ::= { scsiTransportTypes 1 }

scsiTransportSPI     OBJECT-IDENTITY
    STATUS    current
    DESCRIPTION
        "This identity identifies a parallel SCSI transport."
    REFERENCE
        "T10 - SCSI Parallel Interface - 4 (SPI-4)
        - ANSI INCITS 362-2002 [SPI4]"
    ::= { scsiTransportTypes 2 }

scsiTransportFCP     OBJECT-IDENTITY
    STATUS    current
    DESCRIPTION
        "This identity identifies a Fibre Channel Protocol for SCSI,
        Second Version."
    REFERENCE
        "T10 - SCSI Fibre Channel Protocol - 2 (FCP-2)
        - ANSI INCITS 350-2003 [FCP2]"
    ::= { scsiTransportTypes 3 }

scsiTransportSRP     OBJECT-IDENTITY
    STATUS    current
    DESCRIPTION
        "This identity identifies a protocol for transporting SCSI over
        Remote Direct Memory Access (RDMA) interfaces, e.g., InfiniBand
        (tm). "
    REFERENCE
        "T10 - SCSI RDMA Protocol (SRP)
        - ANSI INCITS 365-2002 [SRP]."
    ::= { scsiTransportTypes 4 }

scsiTransportISCSI   OBJECT-IDENTITY
    STATUS    current
    DESCRIPTION

```



```

        "This identity identifies an iSCSI transport."
REFERENCE
    "IETF IPS WG - Internet Small Computer Systems Interface
      (iSCSI) [RFC3720] "
 ::= { scsiTransportTypes 5 }

scsiTransportSBP OBJECT-IDENTITY
    STATUS      current
    DESCRIPTION
        "This identity identifies the Serial Bus Protocol 3."
REFERENCE
    "T10 - Serial Bus Protocol 3 (SBP-3)
      - ANSI INCITS 375-2004 [SBP3]."
 ::= { scsiTransportTypes 6 }

scsiTransportSAS OBJECT-IDENTITY
    STATUS      current
    DESCRIPTION
        "This identity identifies the Serial Attach SCSI Protocol."
REFERENCE
    "T10 - Serial Attached SCSI - 1.1 (SAS - 1.1)
      - #1601-D Rev-10 [SAS-1.1]."
 ::= { scsiTransportTypes 7 }

--***** Instance Table *****
scsiInstanceTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF ScsiInstanceEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A list of SCSI instances present on the system.
        The SCSI instance is the top-level entity, to which everything
        else belongs. An SNMP agent could represent more than one
        instance if it represents either a stack of devices, or virtual
        partitions of a larger device, or a host running multiple SCSI
        implementations from different vendors."
 ::= { scsiGeneral 1 }

scsiInstanceEntry OBJECT-TYPE
    SYNTAX      ScsiInstanceEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry (row) containing management information applicable to
        a particular SCSI instance."
    INDEX { scsiInstIndex }
 ::= { scsiInstanceTable 1 }

```

```

ScsiInstanceEntry ::= SEQUENCE {
    scsiInstIndex          ScsiIndexValue,
    scsiInstAlias          SnmpAdminString,
    scsiInstSoftwareIndex  ScsiHrSWInstalledIndexOrZero,
    scsiInstVendorVersion  SnmpAdminString,
    scsiInstScsiNotificationsEnable TruthValue,
    scsiInstStorageType    StorageType
}

scsiInstIndex OBJECT-TYPE
    SYNTAX      ScsiIndexValue
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This object represents an arbitrary integer used to uniquely
        identify a particular SCSI instance."
    ::= { scsiInstanceEntry 1 }

scsiInstAlias OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE(0..79))
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "This object represents an administrative string, configured by
        the administrator. It can be a zero-length string."
    ::= { scsiInstanceEntry 2 }

scsiInstSoftwareIndex OBJECT-TYPE
    SYNTAX      ScsiHrSWInstalledIndexOrZero
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "If this management instance corresponds to an installed
        software module, then this object's value is the value of the
        hrSWInstalledIndex of that module. If there is no
        correspondence to an installed software module (or no module
        that has an hrSWInstalledIndex value), then the value of this
        object is zero."
    REFERENCE
        "hrSWInstalledIndex is defined in the Host Resources MIB,
        [RFC2790]."
    ::= { scsiInstanceEntry 3 }

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

```

"This object represents a text string set by the manufacturer describing the version of this instance. The format of this string is determined solely by the manufacturer and is for informational purposes only. It is unrelated to the SCSI specification version numbers."

```
::= { scsiInstanceEntry 4 }
```

scsiInstScsiNotificationsEnable OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This object indicates whether notifications defined in this MIB module will be generated."

DEFVAL { true }

```
::= { scsiInstanceEntry 5 }
```

scsiInstStorageType OBJECT-TYPE

SYNTAX StorageType

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This object specifies the memory realization for this SCSI entity.

Specifically, each row in the following tables:

```
    scsiIntrDevTable
    scsiDscTgtTable
    scsiAuthorizedIntrTable
    scsiLunMapTable
```

has a StorageType as specified by the instance of this object that is INDEXed by the same value of scsiInstIndex.

This value of this object is also used to indicate the persistence across reboots of writable values in its row of the scsiInstanceTable.

Conceptual rows having the value 'permanent' need not allow write-access to any columnar objects in the row, nor to any object belonging to a table whose entry is INDEXed by the same value of scsiInstIndex."

DEFVAL { nonVolatile }

```
::= { scsiInstanceEntry 6 }
```

--\*\*\*\*\* Device Table \*\*\*\*\*

scsiDeviceTable OBJECT-TYPE

SYNTAX SEQUENCE OF ScsiDeviceEntry

```

MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION
    "A list of SCSI devices contained in each of the SCSI manageable
    instances that this agent is reporting."
::= { scsiGeneral 2 }

scsiDeviceEntry OBJECT-TYPE
    SYNTAX      ScsiDeviceEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry (row) containing management information applicable to
        a particular SCSI device included in this SCSI manageable
        instance identifiable by the value of scsiInstIndex."
    INDEX {scsiInstIndex, scsiDeviceIndex}
    ::= { scsiDeviceTable 1 }

ScsiDeviceEntry ::= SEQUENCE {
    scsiDeviceIndex      ScsiIndexValue,
    scsiDeviceAlias      SnmpAdminString,
    scsiDeviceRole       BITS,
    scsiDevicePortNumber Unsigned32
}

scsiDeviceIndex OBJECT-TYPE
    SYNTAX      ScsiIndexValue
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This object is an arbitrary integer used to uniquely identify
        a particular device within a particular SCSI instance."
    ::= { scsiDeviceEntry 1 }

scsiDeviceAlias OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE(0..79))
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "This object contains an administrative name for this device.
        If no name is assigned, the value of this object is the
        zero-length string.
        The StorageType of this object is specified by the instance
        of scsiInstStorageType that is INDEXed by the same value of
        scsiInstIndex."
    ::= { scsiDeviceEntry 2 }

scsiDeviceRole OBJECT-TYPE

```

```

SYNTAX      BITS {
    target(0),
    initiator(1)
}
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "This object determines whether this device is acting as a
    SCSI initiator device, or as a SCSI target device, or as both."
 ::= { scsiDeviceEntry 3 }

scsiDevicePortNumber OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object represents the number of ports contained in this
        device."
 ::= { scsiDeviceEntry 4 }

--***** Port Table *****
scsiPortTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF ScsiPortEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A list of SCSI ports for each SCSI device in each instance."
 ::= { scsiGeneral 3 }

scsiPortEntry OBJECT-TYPE
    SYNTAX      ScsiPortEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry (row) containing management information applicable to
        a particular SCSI port of a particular SCSI device in a
        particular SCSI instance."
    INDEX { scsiInstIndex, scsiDeviceIndex, scsiPortIndex }
 ::= { scsiPortTable 1 }

ScsiPortEntry ::= SEQUENCE {
    scsiPortIndex      ScsiIndexValue,
    scsiPortRole        BITS,
    scsiPortTransportPtr RowPointer,
    scsiPortBusyStatuses Counter32
}

```

```

scsiPortIndex OBJECT-TYPE
    SYNTAX      ScsiIndexValue
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An arbitrary integer used to uniquely identify a particular
        port of a given device within a particular SCSI instance."
    ::= { scsiPortEntry 1 }

scsiPortRole OBJECT-TYPE
    SYNTAX      BITS {
        target(0),
        initiator(1)
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object indicates whether this port is acting as a
        SCSI initiator port, or as a SCSI target port or as both."
    ::= { scsiPortEntry 2 }

scsiPortTransportPtr OBJECT-TYPE
    SYNTAX      RowPointer
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object is a pointer to the corresponding row in the
        scsiTransportTable. This row contains information on the
        transport such as transport type and port name."
    ::= { scsiPortEntry 3 }

scsiPortBusyStatuses OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object represents the number of port busy statuses sent or
        received by this port. Note: Initiator ports only receive busy
        status and SCSI target ports only send busy status.
        Discontinuities in the value of this counter can occur at re-
        initialization of the management system."
    ::= { scsiPortEntry 4 }

--***** Table of supported transports *****
scsiTransportTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF ScsiTransportEntry
    MAX-ACCESS  not-accessible

```

```

STATUS      current
DESCRIPTION
    "This table contains the device transport-specific information
    for each transport connected to each device in
    scsiDeviceTable."
 ::= { scsiGeneral 5 }

scsiTransportEntry OBJECT-TYPE
    SYNTAX      ScsiTransportEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry (row) containing parameters applicable to a transport
        used by a particular device of a particular SCSI manageable
        instance."
    INDEX { scsiInstIndex, scsiDeviceIndex, scsiTransportIndex}
 ::= { scsiTransportTable 1 }

ScsiTransportEntry ::= SEQUENCE {
    scsiTransportIndex      ScsiIndexValue,
    scsiTransportType      AutonomousType,
    scsiTransportPointer    RowPointer,
    scsiTransportDevName    ScsiName
}

scsiTransportIndex OBJECT-TYPE
    SYNTAX      ScsiIndexValue
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An arbitrary integer used to uniquely identify a particular
        transport within a given device within a particular SCSI
        instance."
 ::= { scsiTransportEntry 1 }

scsiTransportType OBJECT-TYPE
    SYNTAX      AutonomousType
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object identifies the transport type of this row of the
        transport table.  For example, if this object has the value
        scsiTransportFCP, then the identified transport is FCP."
 ::= { scsiTransportEntry 2 }

scsiTransportPointer OBJECT-TYPE
    SYNTAX      RowPointer
    MAX-ACCESS  read-only

```

```

STATUS      current
DESCRIPTION
    "This object represents a pointer to a conceptual row in a
    'transport' MIB module allowing a manager to get useful
    information for the transport described by this entry.
    For example, if the transport of this device is iSCSI, this
    object will point to the iSCSI Instance of the iSCSI MIB
    module.
    If there is no MIB for this transport, this object has the
    value 0.0."
 ::= { scsiTransportEntry 3 }

scsiTransportDevName OBJECT-TYPE
    SYNTAX      ScsiName
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "This object represents the name of this device in one of the
        format(s) appropriate for this type of transport."
 ::= { scsiTransportEntry 4 }

--***** SCSI Initiator Device Table *****
scsiIntrDevTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF ScsiIntrDevEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "This table contains information for each local SCSI initiator
        device in each instance."
 ::= { scsiInitiatorDevice 1}

scsiIntrDevEntry OBJECT-TYPE
    SYNTAX      ScsiIntrDevEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "An entry (row) containing information applicable to a SCSI
        initiator device within a particular SCSI instance."
    INDEX { scsiInstIndex, scsiDeviceIndex }
 ::= { scsiIntrDevTable 1}

ScsiIntrDevEntry ::= SEQUENCE {
    scsiIntrDevTgtAccessMode    INTEGER,
    scsiIntrDevOutResets        Counter32
}

scsiIntrDevTgtAccessMode OBJECT-TYPE
    SYNTAX      INTEGER {

```



```

        unknown(1),
        autoEnable(2),
        manualEnable(3)
    }
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This object controls whether or not a discovered SCSI target
        device is immediately authorized:
        - autoEnable (2) means that when a SCSI initiator device
        discovers a SCSI target device, it can use it immediately.
        - manualEnable (3) means that the SCSI initiator device
        must wait for an operator to set scsiIntrDscTgtConfigured
        = true before it is authorized.
        The StorageType of this object is specified by the instance
        of scsiInstStorageType that is INDEXed by the same value of
        scsiInstIndex."
 ::= { scsiIntrDevEntry 1 }

scsiIntrDevOutResets OBJECT-TYPE
    SYNTAX        Counter32
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "This object represents the total number of times that this SCSI
        initiator device has issued
        - a LOGICAL UNIT RESET or TARGET RESET task management request,
        or
        - any other SCSI transport protocol-specific action or event
        that causes a Logical Unit Reset or a Hard Reset at one or
        more SCSI target ports ([SAM2] chapters 5.9.6, 5.9.7).
        Discontinuities in the value of this counter can occur at re-
        initialization of the management system."
    REFERENCE
        "SCSI Architecture Model-2 (SAM-2), ANSI INCITS 366-2003,
        T10 Project 1157-D, 12 September 2002
        Chapters 5.9.6 & 5.9.7 [SAM2]"
 ::= { scsiIntrDevEntry 2 }

```

```

-- The following section describes managed objects related to
-- SCSI initiator ports.

```

```

scsiIntrPortTable OBJECT-TYPE
    SYNTAX        SEQUENCE OF ScsiIntrPortEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION

```

"This table contains all the SCSI initiator ports for each local SCSI initiator or target/initiator devices in each SCSI instance."

::= { scsiInitiatorDevice 2 }

scsiIntrPortEntry OBJECT-TYPE

SYNTAX ScsiIntrPortEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry (row) containing information applicable to a particular SCSI initiator port of a particular SCSI device within a SCSI instance."

INDEX { scsiInstIndex, scsiDeviceIndex, scsiPortIndex }

::= { scsiIntrPortTable 1 }

ScsiIntrPortEntry ::= SEQUENCE {

scsiIntrPortName ScsiName,

scsiIntrPortIdentifier ScsiIdentifier,

scsiIntrPortOutCommands Counter32,

scsiIntrPortWrittenMegaBytes Counter32,

scsiIntrPortReadMegaBytes Counter32,

scsiIntrPortHSOutCommands Counter64

}

scsiIntrPortName OBJECT-TYPE

SYNTAX ScsiName

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object represents the name of the port assigned for use by the SCSI protocol. The format will depend on the type of transport this port is using."

::= { scsiIntrPortEntry 1 }

scsiIntrPortIdentifier OBJECT-TYPE

SYNTAX ScsiIdentifier

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object represents the identifier of the port in one of the format(s) appropriate for the type of transport in use."

::= { scsiIntrPortEntry 2 }

scsiIntrPortOutCommands OBJECT-TYPE

SYNTAX Counter32

UNITS "commands"

```
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
    "This object represents the number of commands sent by this
    SCSI initiator port.
    Discontinuities in the value of this counter can occur at re-
    initialization of the management system."
 ::= { scsiIntrPortEntry 3 }

scsiIntrPortWrittenMegaBytes  OBJECT-TYPE
    SYNTAX      Counter32
    UNITS        "Megabytes"
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "This object represents the amount of data in megabytes sent
        by this SCSI initiator port.
        Discontinuities in the value of this counter can occur at re-
        initialization of the management system."
 ::= { scsiIntrPortEntry 4 }

scsiIntrPortReadMegaBytes  OBJECT-TYPE
    SYNTAX      Counter32
    UNITS        "Megabytes"
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "This object represents the amount of data in megabytes
        received by this SCSI initiator port.
        Discontinuities in the value of this counter can occur at re-
        initialization of the management system."
 ::= { scsiIntrPortEntry 5 }

scsiIntrPortHSOutCommands  OBJECT-TYPE
    SYNTAX      Counter64
    UNITS        "commands"
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "This object represents the number of commands sent by this
        SCSI initiator port. This object provides support for systems
        that can quickly generate a large number of commands because
        they run at high speed.
        Discontinuities in the value of this counter can occur at re-
        initialization of the management system."
 ::= { scsiIntrPortEntry 6 }
```

```

--***** Discovered SCSI Target Device group *****
scsiRemoteTgtDev OBJECT IDENTIFIER ::= { scsiInitiatorDevice 3 }

-- SCSI target device discovered or authorized to attach each of the
-- SCSI initiator ports of each SCSI initiator device of each
-- instance.

scsiDscTgtTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF ScsiDscTgtEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "This table includes all the remote (not in the local system)
        SCSI target ports that are authorized to attach to each local
        SCSI initiator port of this SCSI instance."
    ::= { scsiRemoteTgtDev 1 }

scsiDscTgtEntry OBJECT-TYPE
    SYNTAX      ScsiDscTgtEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "Each entry (row) contains information about the SCSI target
        device or port to which this SCSI initiator port (or all SCSI
        initiator ports in the SCSI initiator entry indexed by
        scsiInstIndex, scsiDeviceIndex) will attempt to attach. The
        entry is either for all local ports (if scsiDscTgtIntrPortIndex
        is zero) or only for the specific SCSI initiator port
        identified by scsiDscTgtIntrPortIndex. Note that if an entry in
        this table is deleted, any corresponding entries in the
        scsiDscLunsTable must be deleted as well.
        The StorageType of a row in this table is specified by the
        instance of scsiInstStorageType that is INDEXed by the same
        value of scsiInstIndex."
    INDEX { scsiInstIndex, scsiDeviceIndex, scsiDscTgtIntrPortIndex,
            scsiDscTgtIndex }
    ::= { scsiDscTgtTable 1 }

ScsiDscTgtEntry ::= SEQUENCE {
    scsiDscTgtIntrPortIndex ScsiPortIndexValueOrZero,
    scsiDscTgtIndex         ScsiIndexValue,
    scsiDscTgtDevOrPort     ScsiDeviceOrPort,
    scsiDscTgtName          ScsiName,
    scsiDscTgtConfigured    TruthValue,
    scsiDscTgtDiscovered    TruthValue,
    scsiDscTgtInCommands    Counter32,
    scsiDscTgtWrittenMegaBytes Counter32,
    scsiDscTgtReadMegaBytes Counter32,

```

```

        scsiDscTgtHSInCommands Counter64,
        scsiDscTgtLastCreation TimeStamp,
        scsiDscTgtRowStatus RowStatus
    }

scsiDscTgtIntrPortIndex OBJECT-TYPE
    SYNTAX      ScsiPortIndexValueOrZero
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This object relates to a particular local device within a
        particular SCSI instance and specifies
        - the index of the local SCSI initiator port,
        - or zero, if this entry refers to the local device and
        therefore refers to all the local SCSI initiator ports."
    ::= { scsiDscTgtEntry 1 }

scsiDscTgtIndex OBJECT-TYPE
    SYNTAX      ScsiIndexValue
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This object is an arbitrary integer used to uniquely identify
        a particular SCSI target device either discovered by, or
        configured for use with, one or more ports scsiDscTgtName of
        a particular device within a particular SCSI instance."
    ::= { scsiDscTgtEntry 2 }

scsiDscTgtDevOrPort OBJECT-TYPE
    SYNTAX      ScsiDeviceOrPort
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This object indicates whether this entry describes a
        configured SCSI target device name (and applies to all ports
        on the identified SCSI target device) or an individual SCSI
        target port."
    ::= { scsiDscTgtEntry 3 }

scsiDscTgtName OBJECT-TYPE
    SYNTAX      ScsiName
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This object represents the name of this configured or
        discovered SCSI target device or port depending on the value
        of scsiDscTgtDevOrPort."
    ::= { scsiDscTgtEntry 4 }

```

## scsiDscTgtConfigured OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-create

STATUS current

## DESCRIPTION

"This object means

-true(1): this entry has been configured by an administrator.

-false(2): this entry has been added from a discovery mechanism (e.g., SendTargets, SLP, iSNS).

An administrator can modify this value from false to true."

DEFVAL { true }

::= { scsiDscTgtEntry 5 }

## scsiDscTgtDiscovered OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"This object means

-true(1): this entry has been discovered by the SCSI instance as result of an automatic discovery process.

-false(2): this entry has been added by manual configuration. This entry is read-only because an administrator cannot change it.

Note that it is an implementation decision to determine how long to retain a row with configured=false, such as when the SCSI target device is no longer visible/accessible to the local SCSI initiator device."

::= { scsiDscTgtEntry 6 }

## scsiDscTgtInCommands OBJECT-TYPE

SYNTAX Counter32

UNITS "commands"

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"This object represents the number of commands received from this SCSI target port or device.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of scsiDscTgtLastCreation."

::= { scsiDscTgtEntry 7 }

## scsiDscTgtWrittenMegaBytes OBJECT-TYPE

SYNTAX Counter32

UNITS "Megabytes"

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"This object represents the amount of megabytes of data sent as the result of WRITE commands to this SCSI target port or device. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of scsiDscTgtLastCreation."

```
::= { scsiDscTgtEntry 8 }
```

## scsiDscTgtReadMegaBytes OBJECT-TYPE

SYNTAX Counter32  
 UNITS "Megabytes"  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION

"This object represents the amount of megabytes received as the result of READ commands to this SCSI target port or device. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of scsiDscTgtLastCreation."

```
::= { scsiDscTgtEntry 9 }
```

## scsiDscTgtHSInCommands OBJECT-TYPE

SYNTAX Counter64  
 UNITS "commands"  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION

"This object represents the number of commands received by this SCSI target port or device. This object provides support for system that can quickly generate a large number of commands because they run at high speed. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of scsiDscTgtLastCreation."

```
::= { scsiDscTgtEntry 10 }
```

## scsiDscTgtLastCreation OBJECT-TYPE

SYNTAX TimeStamp  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION

"This object represents the value of sysUpTime when this row was created."

```
::= { scsiDscTgtEntry 11 }
```

## scsiDscTgtRowStatus OBJECT-TYPE

SYNTAX RowStatus  
 MAX-ACCESS read-create

STATUS           current  
DESCRIPTION

"This object allows an administrator to configure dynamically a new entry in this table via SNMP or eventually delete it. An administrator is not allowed to delete an entry for which the value of the object scsiIntrDscTgtDiscovered is equal to true.

Note that when an entry in this table is deleted, then any corresponding entries in the scsiDscLunTable must also be automatically deleted.

A newly created row cannot be made active until a value has been set for scsiDscTgtName. In this case, the value of the corresponding instance of the scsiDscTgtRowStatus column will stay 'notReady'.

The RowStatus TC [RFC2579] requires that this DESCRIPTION clause states under which circumstances other objects in this row can be modified:

The value of this object has no effect on whether other objects in this conceptual row can be modified."

::= { scsiDscTgtEntry 12 }

--\*\*\*\*\* LUNs discovered \*\*\*\*\*

scsiDscLunTable OBJECT-TYPE

SYNTAX           SEQUENCE OF ScsiDscLunEntry  
MAX-ACCESS       not-accessible  
STATUS           current

DESCRIPTION

"This table includes all the remote (not in the local system) logical unit numbers (LUNs) discovered via each local SCSI initiator port of each local device within a particular SCSI instance."

::= { scsiRemoteTgtDev 2 }

scsiDscLunEntry OBJECT-TYPE

SYNTAX           ScsiDscLunEntry  
MAX-ACCESS       not-accessible  
STATUS           current

DESCRIPTION

"An entry (row) represents a discovered LUN at a particular SCSI target device (scsiDscTgtIndex), where the LUN was discovered by a particular local SCSI initiator device within a particular SCSI instance, possibly via a particular local SCSI initiator port.

Note that when an entry in the scsiDscTgtTable is deleted, all corresponding entries in this table should automatically be deleted."



```

    INDEX { scsiInstIndex, scsiDeviceIndex, scsiDscTgtIntrPortIndex,
             scsiDscTgtIndex, scsiDscLunIndex }
 ::= { scsiDscLunTable 1 }

ScsiDscLunEntry ::= SEQUENCE {
    scsiDscLunIndex    ScsiIndexValue,
    scsiDscLunLun      ScsiLUN
}

scsiDscLunIndex OBJECT-TYPE
    SYNTAX      ScsiIndexValue
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This object is an arbitrary integer used to uniquely identify
        a particular LUN discovered by a particular SCSI initiator port
        or a particular SCSI initiator device within a particular SCSI
        instance.
        Entries in the scsiDscLunIdTable are associated with a LUN by
        having the value of this object in their INDEX."
 ::= { scsiDscLunEntry 1 }

scsiDscLunLun OBJECT-TYPE
    SYNTAX      ScsiLUN
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object contains the Logical Unit Number (LUN) of the
        discovered logical unit."
 ::= { scsiDscLunEntry 2 }

--***** LU Identifiers discovered *****
scsiDscLunIdTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF ScsiDscLunIdEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table includes all the known LU identifiers of the remote
        (not in the local system) logical units discovered via each
        local SCSI initiator port or device of this SCSI instance."
 ::= { scsiRemoteTgtDev 3 }

scsiDscLunIdEntry OBJECT-TYPE
    SYNTAX      ScsiDscLunIdEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION

```

"An entry (row) represents the LU identifier of a discovered LUN at a particular SCSI target device (scsiDscTgtIndex), where the LUN was discovered by a particular local SCSI initiator device within a particular SCSI instance, possibly via a particular local SCSI initiator port."

```

INDEX { scsiInstIndex, scsiDeviceIndex, scsiDscTgtIntrPortIndex,
        scsiDscTgtIndex, scsiDscLunIndex, scsiDscLunIdIndex }
 ::= { scsiDscLunIdTable 1 }

ScsiDscLunIdEntry ::= SEQUENCE {
    scsiDscLunIdIndex      ScsiIndexValue,
    scsiDscLunIdCodeSet    ScsiIdCodeSet,
    scsiDscLunIdAssociation ScsiIdAssociation,
    scsiDscLunIdType       ScsiIdType,
    scsiDscLunIdValue      ScsiIdValue
}

scsiDscLunIdIndex OBJECT-TYPE
    SYNTAX      ScsiIndexValue
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This object is an arbitrary integer used to uniquely identify
        a particular LUN identifier discovered by each SCSI initiator
        device or particular SCSI initiator port within a particular
        SCSI instance."
    ::= { scsiDscLunIdEntry 1 }

scsiDscLunIdCodeSet OBJECT-TYPE
    SYNTAX      ScsiIdCodeSet
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object specifies the code set in use with this
        identifier. The value is represented in the same format as
        is contained in the identifier's Identification Descriptor
        within the logical unit's Device Identification Page."
    REFERENCE
        "ANSI - SCSI Primary Commands - 2 (SPC-2),
        ANSI INCITS 351-2001, 11 July 2001 Chapter 8: section 8.4.4,
        Vital Product Data Parameters [SPC2]"
    ::= { scsiDscLunIdEntry 2 }

scsiDscLunIdAssociation OBJECT-TYPE
    SYNTAX      ScsiIdAssociation
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION

```

"This object specifies what the identifier is associated with (e.g., with the addressed physical/logical device or with a particular port). The value is represented in the same format as is contained in the identifier's Identification Descriptor within the logical unit's Device Identification Page."

## REFERENCE

"ANSI - SCSI Primary Commands - 2 (SPC-2),  
ANSI INCITS 351-2001, 11 July 2001 Chapter 8: [section 8.4.4](#),  
Vital Product Data Parameters [[SPC2](#)]"

::= { scsiDscLunIdEntry 3 }

scsiDscLunIdType OBJECT-TYPE

SYNTAX ScsiIdType  
MAX-ACCESS read-only  
STATUS current

## DESCRIPTION

"This object specifies the type of the identifier.  
The value is represented in the same format as is contained in the identifier's Identification Descriptor within the logical unit's Device Identification Page."

## REFERENCE

"ANSI - SCSI Primary Commands - 2 (SPC-2),  
ANSI INCITS 351-2001, 11 July 2001 Chapter 8: [section 8.4.4](#),  
Vital Product Data Parameters [[SPC2](#)]"

::= { scsiDscLunIdEntry 4 }

scsiDscLunIdValue OBJECT-TYPE

SYNTAX ScsiIdValue  
MAX-ACCESS read-only  
STATUS current

## DESCRIPTION

"This object represents the actual value of this identifier.  
The format is defined by the objects scsiDscLunIdCodeSet, scsiDscLunIdAssociation, scsiDscLunIdType.  
The value is represented in the same format as is contained in the identifier's Identification Descriptor within the logical unit's Device Identification Page."

## REFERENCE

"ANSI - SCSI Primary Commands - 2 (SPC-2),  
ANSI INCITS 351-2001, 11 July 2001 Chapter 8: [section 8.4.4](#),  
Vital Product Data Parameters [[SPC2](#)]"

::= { scsiDscLunIdEntry 5 }

--\*\*\*\*\* Table of SCSI Target Device Attached to local SCSI  
--\*\*\*\*\* Initiator Ports

scsiAttTgtPortTable OBJECT-TYPE

SYNTAX SEQUENCE OF ScsiAttTgtPortEntry  
MAX-ACCESS not-accessible

```

STATUS          current
DESCRIPTION
    "This table includes all the remote (not in the local system)
    SCSI target ports that are currently attached to each local
    SCSI initiator port of this SCSI instance."
 ::= { scsiRemoteTgtDev 4 }

scsiAttTgtPortEntry OBJECT-TYPE
    SYNTAX      ScsiAttTgtPortEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "An entry (row) represents a remote SCSI target port
        (scsiAttTgtPortIndex) currently attached to a particular
        SCSI initiator port (scsiPortIndex) of a particular SCSI
        initiator device within a particular SCSI instance."
    INDEX { scsiInstIndex, scsiDeviceIndex, scsiPortIndex,
            scsiAttTgtPortIndex }
 ::= { scsiAttTgtPortTable 1 }

ScsiAttTgtPortEntry ::= SEQUENCE {
    scsiAttTgtPortIndex      ScsiIndexValue,
    scsiAttTgtPortDscTgtIdx  ScsiIndexValueOrZero,
    scsiAttTgtPortName       ScsiName,
    scsiAttTgtPortIdentifier  ScsiIdentifier
}

scsiAttTgtPortIndex OBJECT-TYPE
    SYNTAX      ScsiIndexValue
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "An arbitrary integer used to uniquely identify a particular
        SCSI target currently attached to a particular SCSI initiator
        port of a particular SCSI initiator device within a particular
        SCSI instance."
 ::= { scsiAttTgtPortEntry 1 }

scsiAttTgtPortDscTgtIdx OBJECT-TYPE
    SYNTAX      ScsiIndexValueOrZero
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "This object contains the value of the scsiDscTgtIntrPortIndex
        index variable for the row in the scsiDscTgtTable representing
        this currently attached SCSI target port.  If the currently
        attached SCSI target port is not represented in the
        scsiDscTgtTable, then the value of this object is zero."

```

```

 ::= { scsiAttTgtPortEntry 2 }

scsiAttTgtPortName OBJECT-TYPE
    SYNTAX      ScsiName
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "This object contains the name of the attached SCSI target
        port."
 ::= { scsiAttTgtPortEntry 3 }

scsiAttTgtPortIdentifier OBJECT-TYPE
    SYNTAX      ScsiIdentifier
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "This object contains the identifier of the attached SCSI
        target port."
 ::= { scsiAttTgtPortEntry 4 }

-- *****
-- ***** Table of SCSI Target devices
--
scsiTgtDevTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF ScsiTgtDevEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "This table contains information about each local SCSI target
        device."
 ::= { scsiTargetDevice 1 }

scsiTgtDevEntry OBJECT-TYPE
    SYNTAX      ScsiTgtDevEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "An entry (row) containing information applicable to a
        particular local SCSI target device within a particular SCSI
        instance."
    INDEX { scsiInstIndex, scsiDeviceIndex }
 ::= { scsiTgtDevTable 1 }

ScsiTgtDevEntry ::= SEQUENCE {
    scsiTgtDevNumberOfLUs      Gauge32,
    scsiTgtDeviceStatus        INTEGER,
    scsiTgtDevNonAccessibleLUs Gauge32,
    scsiTgtDevResets           Counter32

```

```

}

scsiTgtDevNumberOfLUs OBJECT-TYPE
    SYNTAX      Gauge32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "This object is the number of logical units accessible via this
        local SCSI target device."
    ::= { scsiTgtDevEntry 1 }

scsiTgtDeviceStatus OBJECT-TYPE
    SYNTAX      INTEGER {
        unknown(1),
        available(2),
        broken(3),
        readying(4),
        abnormal(5),
        nonAddrFailure(6),
        nonAddrFailReadying(7),
        nonAddrFailAbnormal(8)
    }
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "This object represents the status of this SCSI device,
        summarizing the state of both the addressable devices (i.e.,
        the logical units) and the non-addressable devices within this
        SCSI device:
        - unknown(1): This value is used when the status cannot be
          determined
        - available(2): All addressable and non-addressable
          devices within the SCSI device are fully operational (i.e.,
          no logical units have an abnormal status).
        - broken(3): The SCSI device is not operational and cannot
          be made operational without external intervention.
        - readying(4): One or more logical units within the SCSI
          device are being initialized and access to the SCSI device
          is temporarily limited (i.e., one or more of the logical
          units have a readying status).
        - abnormal(5): One or more addressable devices within the
          SCSI device are indicating a status other than available;
          nevertheless, the SCSI device is operational (i.e., one or
          more of the logical units have an abnormal status).
        - nonAddrFailure(6): One or more non-addressable devices
          within the SCSI device have failed; nevertheless, the SCSI
          device is operational (i.e., no logical units have an
          abnormal or readying status)."

```

- nonAddrFailReady(7): One or more non-addressable devices within the SCSI device have failed; nevertheless, one or more logical units within the SCSI device are being initialized and access to the SCSI device is temporarily limited.

- nonAddrFailAbnormal(8): One or more non-addressable devices within the SCSI device have failed and one or more addressable devices within the SCSI device are indicating a status other than available; however, the SCSI device is operational.

"

#### REFERENCE

"SCSI Controller Commands-2 (SCC-2) ANSI INCITS 318-1998  
6.3.1.8 REPORT STATES service action [[SCC2](#)]"

::= { scsiTgtDevEntry 2 }

#### scsiTgtDevNonAccessibleLUs OBJECT-TYPE

SYNTAX Gauge32  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION

"This object is the number of logical units existing but not currently accessible via this local SCSI target device."

::= { scsiTgtDevEntry 3 }

#### scsiTgtDevResets OBJECT-TYPE

SYNTAX Counter32  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION

"This object counts the number of hard resets encountered by this SCSI target device.  
Discontinuities in the value of this counter can occur at re-initialization of the management system."

#### REFERENCE

"SCSI Architecture Model-2 (SAM-2), ANSI INCITS 366-2003,  
T10 Project 1157-D, 12 September 2002 - Chapter 5.9.7 [[SAM2](#)]"

::= { scsiTgtDevEntry 4 }

#### --\*\*\*\*\* SCSI Target Port Table \*\*\*\*\*

#### scsiTgtPortTable OBJECT-TYPE

SYNTAX SEQUENCE OF ScsiTgtPortEntry  
MAX-ACCESS not-accessible  
STATUS current  
DESCRIPTION

"This table includes all the local SCSI target ports of all the local SCSI target devices."

```

 ::= { scsiTargetDevice 2 }

scsiTgtPortEntry OBJECT-TYPE
    SYNTAX      ScsiTgtPortEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "An entry (row) containing information applicable to a
        particular local SCSI target port of a particular local SCSI
        target device within a particular SCSI instance."
    INDEX { scsiInstIndex, scsiDeviceIndex, scsiPortIndex}
 ::= { scsiTgtPortTable 1 }

ScsiTgtPortEntry ::= SEQUENCE {
    scsiTgtPortName          ScsiName,
    scsiTgtPortIdentifier    ScsiIdentifier,
    scsiTgtPortInCommands    Counter32,
    scsiTgtPortWrittenMegaBytes Counter32,
    scsiTgtPortReadMegaBytes Counter32,
    scsiTgtPortHSInCommands Counter64
}

scsiTgtPortName OBJECT-TYPE
    SYNTAX      ScsiName
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "This object represents the name of the port assigned for use
        in the SCSI protocol."
 ::= { scsiTgtPortEntry 1 }

scsiTgtPortIdentifier OBJECT-TYPE
    SYNTAX      ScsiIdentifier
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "This object represents the identifier of the port in one of
        the format(s) appropriate for the type of transport."
 ::= { scsiTgtPortEntry 2 }

scsiTgtPortInCommands OBJECT-TYPE
    SYNTAX      Counter32
    UNITS       "commands"
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "This object represents the number of commands received by this
        SCSI target port."

```



Discontinuities in the value of this counter can occur at re-initialization of the management system."

```
::= { scsiTgtPortEntry 3 }
```

scsiTgtPortWrittenMegaBytes OBJECT-TYPE

```
SYNTAX      Counter32
UNITS       "Megabytes"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "This object represents the amount of data written in megabytes
    by this SCSI target port.
    Discontinuities in the value of this counter can occur at re-
    initialization of the management system."
::= { scsiTgtPortEntry 4 }
```

scsiTgtPortReadMegaBytes OBJECT-TYPE

```
SYNTAX      Counter32
UNITS       "Megabytes"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "This object represents the amount of data read in megabytes by
    this SCSI target port.
    Discontinuities in the value of this counter can occur at re-
    initialization of the management system."
::= { scsiTgtPortEntry 5 }
```

scsiTgtPortHSInCommands OBJECT-TYPE

```
SYNTAX      Counter64
UNITS       "commands"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "This object represents the number of commands received.  This
    object provides support for systems that can quickly generate a
    large number of commands because they run at high speed.
    Discontinuities in the value of this counter can occur at re-
    initialization of the management system."
::= { scsiTgtPortEntry 6 }
```

scsiRemoteIntrDev OBJECT IDENTIFIER ::= { scsiTargetDevice 3 }

```
-- The scsiAuthorizedIntrTable contains the list of remote initiator
-- ports that are authorized to be attached to specific SCSI target
-- ports and on which an administrator would like to keep permanent
-- information and long term statistics even when not currently
-- attached.
```

## scsiAuthorizedIntrTable OBJECT-TYPE

SYNTAX SEQUENCE OF ScsiAuthorizedIntrEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

"This table includes all the authorized SCSI initiator devices or ports that may attach a SCSI target device (ScsiAuthIntrTgtPortIndex = 0) or port (ScsiAuthIntrTgtPortIndex different than 0) of the local SCSI instance. Statistics are kept for each such authorization; thus, the authorizations should be configured in the manner that will cause the desired set of statistics to be collected and that will determine the correct LUN map."

::= { scsiRemoteIntrDev 1 }

## scsiAuthorizedIntrEntry OBJECT-TYPE

SYNTAX ScsiAuthorizedIntrEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

"An entry (row) represents a remote SCSI initiator port or remote SCSI initiator device that may attach to the local SCSI target port or device within a particular SCSI instance. The StorageType of a row in this table is specified by the instance of scsiInstStorageType that is INDEXed by the same value of scsiInstIndex."

INDEX { scsiInstIndex, scsiDeviceIndex, scsiAuthIntrTgtPortIndex, scsiAuthIntrIndex }

::= { scsiAuthorizedIntrTable 1 }

## ScsiAuthorizedIntrEntry ::= SEQUENCE {

scsiAuthIntrTgtPortIndex ScsiPortIndexValueOrZero,

scsiAuthIntrIndex ScsiIndexValue,

scsiAuthIntrDevOrPort ScsiDeviceOrPort,

scsiAuthIntrName ScsiName,

scsiAuthIntrLunMapIndex ScsiIndexValueOrZero,

scsiAuthIntrAttachedTimes Counter32,

scsiAuthIntrOutCommands Counter32,

scsiAuthIntrReadMegaBytes Counter32,

scsiAuthIntrWrittenMegaBytes Counter32,

scsiAuthIntrHSOutCommands Counter64,

scsiAuthIntrLastCreation TimeStamp,

scsiAuthIntrRowStatus RowStatus

}

## scsiAuthIntrTgtPortIndex OBJECT-TYPE

SYNTAX ScsiPortIndexValueOrZero

```
MAX-ACCESS    not-accessible
STATUS        current
DESCRIPTION
    "This object contains either the index of the port or zero, to
    indicate any port, on the particular local SCSI target device."
 ::= { scsiAuthorizedIntrEntry 1 }

scsiAuthIntrIndex OBJECT-TYPE
    SYNTAX      ScsiIndexValue
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This object is an arbitrary integer used to uniquely identify
        a SCSI initiator device or port that is authorized to attach
        to a particular local SCSI target device or port of a particular
        SCSI instance."
 ::= { scsiAuthorizedIntrEntry 2 }

scsiAuthIntrDevOrPort OBJECT-TYPE
    SYNTAX      ScsiDeviceOrPort
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This object specifies whether this entry refers to a remote
        SCSI initiator port or to a SCSI initiator device.
        A value of device(1) means that the authorized remote initiator
        is a SCSI initiator device and includes all of its ports.
        A value of port(2) means that the authorized remote initiator
        is a SCSI initiator port."
 ::= { scsiAuthorizedIntrEntry 3 }

scsiAuthIntrName OBJECT-TYPE
    SYNTAX      ScsiName
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This object represents the name of the remote SCSI initiator
        device or port authorized by this row."
 ::= { scsiAuthorizedIntrEntry 4 }

scsiAuthIntrLunMapIndex OBJECT-TYPE
    SYNTAX      ScsiIndexValueOrZero
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This object identifies the set of entries in the
        scsiLunMapTable for which scsiLunMapIndex has the same value as
        the value of this object. The identified set of entries
```

constitutes the LUN map to be used for accessing logical units when the remote SCSI initiator port or device corresponding to this entry is attached to any local SCSI target port or device corresponding to this entry.

Note that this object has a value of zero if this entry should use the default LUN map."

```
::= { scsiAuthorizedIntrEntry 5 }
```

scsiAuthIntrAttachedTimes OBJECT-TYPE

SYNTAX Counter32

UNITS "Times"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object indicates the number of times that this remote SCSI initiator device or port has transitioned from unattached to attached to this local SCSI target device or port.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of scsiAuthIntrLastCreation."

```
::= { scsiAuthorizedIntrEntry 6 }
```

scsiAuthIntrOutCommands OBJECT-TYPE

SYNTAX Counter32

UNITS "commands"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object indicates the number of commands that the remote SCSI initiator device or port corresponding to this entry has sent to the local SCSI target device or port corresponding to this entry.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of scsiAuthIntrLastCreation."

```
::= { scsiAuthorizedIntrEntry 7 }
```

scsiAuthIntrReadMegaBytes OBJECT-TYPE

SYNTAX Counter32

UNITS "Megabytes"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object indicates the amount of data in megabytes that the remote SCSI initiator device or port corresponding to this entry has read from the local SCSI target device or port corresponding to this entry.

Discontinuities in the value of this counter can occur at re-

```

        initialization of the management system, and at other times as
        indicated by the value of scsiAuthIntrLastCreation."
 ::= { scsiAuthorizedIntrEntry 8 }

scsiAuthIntrWrittenMegaBytes OBJECT-TYPE
    SYNTAX      Counter32
    UNITS       "Megabytes"
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "This object indicates the amount of data in megabytes that the
        remote SCSI initiator device or port corresponding to this
        entry has written to the local SCSI target device or port
        corresponding to this entry.
        Discontinuities in the value of this counter can occur at re-
        initialization of the management system, and at other times as
        indicated by the value of scsiAuthIntrLastCreation."
 ::= { scsiAuthorizedIntrEntry 9}

scsiAuthIntrHSOutCommands OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "commands"
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "This object represents the number of commands sent by the
        remote SCSI initiator device or port corresponding to this
        entry to the local SCSI target device or port corresponding to
        this entry. This object provides support for systems that can
        quickly generate a large number of commands because they run at
        high speed.
        Discontinuities in the value of this counter can occur at re-
        initialization of the management system, and at other times as
        indicated by the value of scsiAuthIntrLastCreation."
 ::= { scsiAuthorizedIntrEntry 10 }

scsiAuthIntrLastCreation OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "This object indicates the value of sysUpTime when this row was
        last created."
 ::= { scsiAuthorizedIntrEntry 11 }

scsiAuthIntrRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS   read-create

```

```

STATUS          current
DESCRIPTION
    "This object allows an administrator to create or delete this
    entry.
    A newly created row cannot be made active until a value has
    been set for scsiAuthIntrName.  In this case, the value of the
    corresponding instance of the scsiAuthIntrRowStatus column will
    stay 'notReady'.
    The RowStatus TC [RFC2579] requires that this DESCRIPTION
    clause states under which circumstances other objects in this
    row can be modified:
    The value of this object has no effect on whether other objects
    in this conceptual row can be modified."
 ::= { scsiAuthorizedIntrEntry 12 }

-- Table of SCSI initiator devices or ports attached to local
-- SCSI target ports
--

scsiAttIntrPortTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF ScsiAttIntrPortEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "This table includes all the remote SCSI initiator ports that
        are currently attached to a local SCSI target port of all local
        devices within all SCSI instances."
 ::= { scsiRemoteIntrDev 2 }

scsiAttIntrPortEntry OBJECT-TYPE
    SYNTAX      ScsiAttIntrPortEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "An entry (row) represents a remote SCSI initiator port
        currently attached to a particular local SCSI target port of a
        particular SCSI target device of a particular SCSI instance."
    INDEX { scsiInstIndex, scsiDeviceIndex, scsiPortIndex,
            scsiAttIntrPortIndex }
 ::= { scsiAttIntrPortTable 1 }

ScsiAttIntrPortEntry ::= SEQUENCE {
    scsiAttIntrPortIndex      ScsiIndexValue,
    scsiAttIntrPortAuthIntrIdx ScsiIndexValueOrZero,
    scsiAttIntrPortName       ScsiName,
    scsiAttIntrPortIdentifier ScsiIdentifier
}

```

## scsiAttIntrPortIndex OBJECT-TYPE

SYNTAX ScsiIndexValue

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

"This object represents an arbitrary integer used to uniquely identify a particular attached remote initiator port to a particular SCSI target port within a particular SCSI target device within a particular SCSI instance."

::= { scsiAttIntrPortEntry 1 }

## scsiAttIntrPortAuthIntrIdx OBJECT-TYPE

SYNTAX ScsiIndexValueOrZero

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"This object is the corresponding index in the scsiAuthorizedIntrTable for this current attached remote SCSI initiator device or zero if this remote attached SCSI initiator device is not configured in that table."

::= { scsiAttIntrPortEntry 2 }

## scsiAttIntrPortName OBJECT-TYPE

SYNTAX ScsiName

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"This object represents the name of the remote SCSI initiator device attached to this local SCSI target port."

::= { scsiAttIntrPortEntry 3 }

## scsiAttIntrPortIdentifier OBJECT-TYPE

SYNTAX ScsiIdentifier

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"This object represents the identifier of the remote SCSI initiator device attached to this local SCSI target port."

::= { scsiAttIntrPortEntry 4 }

--\*\*\*\*\* Managed Objects regarding logical units \*\*\*\*\*

## scsiLuTable OBJECT-TYPE

SYNTAX SEQUENCE OF ScsiLuEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

"This table contains the logical units exposed by local SCSI target devices."

It includes attributes for the World Wide Name (WWN), scsiLuVendorId, scsiLuProductId, and scsiLuRevisionId, which may also appear in the scsiLuIdTable. If an implementation exposes a WWN as a LuIdTable entry, it must match the scsiLuWwnName in this table. If an implementation exposes a (vendor, product, revision) identifier as an LuIdTable entry, each of these fields must match the scsiLuVendorId, scsiLuProductId, and scsiLuRevisionId attributes in this table."

```
::= { scsiLogicalUnit 1 }
```

scsiLuEntry OBJECT-TYPE

SYNTAX ScsiLuEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry (row) contains information applicable to a particular logical unit of a particular local SCSI target device within a particular SCSI instance."

INDEX { scsiInstIndex, scsiDeviceIndex, scsiLuIndex }

```
::= { scsiLuTable 1 }
```

ScsiLuEntry ::= SEQUENCE {

```
    scsiLuIndex          ScsiIndexValue,
    scsiLuDefaultLun     ScsiLUN,
    scsiLuWwnName        ScsiLuNameOrZero,
    scsiLuVendorId       SnmpAdminString,
    scsiLuProductId      SnmpAdminString,
    scsiLuRevisionId     SnmpAdminString,
    scsiLuPeripheralType Unsigned32,
    scsiLuStatus         INTEGER,
    scsiLuState          BITS,
    scsiLuInCommands     Counter32,
    scsiLuReadMegaBytes  Counter32,
    scsiLuWrittenMegaBytes Counter32,
    scsiLuInResets       Counter32,
    scsiLuOutTaskSetFullStatus Counter32,
    scsiLuHSInCommands   Counter64,
    scsiLuLastCreation    TimeStamp
}
```

scsiLuIndex OBJECT-TYPE

SYNTAX ScsiIndexValue

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This object represents an arbitrary integer used to uniquely identify a particular logical unit within a particular SCSI target device within a particular SCSI instance."



```
::= { scsiLuEntry 1 }
```

scsiLuDefaultLun OBJECT-TYPE

SYNTAX ScsiLUN

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object represents the default Logical Unit Number (LUN) for this logical unit; if a SCSI initiator device has not been configured to view this logical unit via an entry in the ScsiLunMapTable, the LU will be visible as scsiLuDefaultLun. If this logical unit does not have a default LUN, it will only be visible if specified via the ScsiLunMapTable, and this object will contain a zero-length string."

```
::= { scsiLuEntry 2 }
```

scsiLuWwnName OBJECT-TYPE

SYNTAX ScsiLuNameOrZero

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object represents the World Wide Name of this LU that is the device identifier of the Vital Product Data (VPD) page name; if there is no WWN for this LU, this object will contain a zero-length string. If there is more than one identifier, they will be listed in the scsiLuIdTable and this object will contain a zero-length string."

```
::= { scsiLuEntry 3 }
```

scsiLuVendorId OBJECT-TYPE

SYNTAX SnmpAdminString

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object represents a string identifying the vendor of this LU as reported in the Standard INQUIRY data."

```
::= { scsiLuEntry 4 }
```

scsiLuProductId OBJECT-TYPE

SYNTAX SnmpAdminString

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object represents a string identifying the product for this LU as reported in the Standard INQUIRY data."

```
::= { scsiLuEntry 5 }
```

scsiLuRevisionId OBJECT-TYPE

SYNTAX            SnmpAdminString  
MAX-ACCESS       read-only  
STATUS            current  
DESCRIPTION  
    "This object represents a string defining the product revision  
    of this LU as reported in the Standard INQUIRY data."  
 ::= { scsiLuEntry 6 }

scsiLuPeripheralType OBJECT-TYPE

SYNTAX            Unsigned32  
MAX-ACCESS       read-only  
STATUS            current  
DESCRIPTION  
    "This object is the value returned by SCSI Standard INQUIRY  
    data. It can be: direct-access device, sequential-access  
    device, printer, communication device and so on.  
    The values that can be returned here are defined in SCSI  
    Primary Commands -2."  
REFERENCE  
    "ANSI - SCSI Primary Commands - 2 (SPC-2),  
    ANSI INCITS 351-2001,11 July 2001 [[SPC2](#)]- Table 48."  
 ::= { scsiLuEntry 7 }

scsiLuStatus OBJECT-TYPE

SYNTAX            INTEGER {  
    unknown(1),  
    available(2),  
    notAvailable(3),  
    broken(4),  
    readying(5),  
    abnormal(6)  
}

MAX-ACCESS       read-only  
STATUS            current

DESCRIPTION

"This object represents the status of this logical unit:  
- unknown(1): The status of this logical unit cannot be  
determined.  
- available(2): The logical unit is fully operational (i.e.,  
accepts media access SCSI commands and has no state  
information to report).  
- notAvailable(3): The logical unit is capable of being  
supported but is not available (i.e., no logical unit is  
currently present or the logical unit is present but not  
configured for use).  
- broken(4): The logical unit has failed and cannot respond  
to SCSI commands.  
- readying(5): The logical unit is being initialized and

access is temporarily limited.

- abnormal(6): The logical unit has state information available that indicates it is operating with limits. The scsiLuState indicates what those limits are.

"

#### REFERENCE

"SCSI Controller Commands-2 (SCC-2) ANSI INCITS 318-1998

6.3.1.8 REPORT STATES service action [[SCC2](#)]"

::= { scsiLuEntry 8 }

#### scsiLuState OBJECT-TYPE

```
SYNTAX      BITS {
    dataLost(0),
    dynamicReconfigurationInProgress(1),
    exposed(2),
    fractionallyExposed(3),
    partiallyExposed(4),
    protectedRebuild(5),
    protectionDisabled(6),
    rebuild(7),
    recalculate(8),
    spareInUse(9),
    verifyInProgress(10)
}
```

MAX-ACCESS read-only

STATUS current

#### DESCRIPTION

"This object represents the state of a logical unit and its meaning according to the bit position:

- 0 Data lost: Within the logical unit data has been lost.
- 1 Dynamic reconfiguration in progress: The logical unit is being reconfigured. In this state all data is still protected.
- 2 Exposed: Within the logical unit data is not protected. In this state all data is still valid; however, loss of data or data availability is unavoidable in the event of a failure.
- 3 Fractionally exposed: Within the logical unit part of the data is not protected. In this state all data is still valid; however, a failure may cause a loss of data or a loss of data availability.
- 4 Partially exposed: Within the logical unit one or more underlying storage devices have failed. In this state all data is still protected.
- 5 Protected rebuild: The logical unit is in the process of a rebuild operation. In this state all data is protected.
- 6 Protection disabled: Within the logical unit the data

protection method has been disabled.

In this state all data is still valid; however, loss of data or data availability is unavoidable in the event of a failure.

7 Rebuild: The data protection method is in the process of rebuilding data. In this state data is not protected.

8 Recalculate: The logical unit is in the process of a recalculate operation.

9 Spare in use: Within the logical unit a storage device in full or part is being used to store data. In this state all data is still protected.

10 Verify in progress: Within the logical unit data is being verified."

#### REFERENCE

"SCSI Controller Commands-2 (SCC-2) ANSI INCITS 318-1998  
6.3.1.8 REPORT STATES service action [SCC2]"

```
::= { scsiLuEntry 9 }
```

#### scsiLuInCommands OBJECT-TYPE

SYNTAX Counter32

UNITS "commands"

MAX-ACCESS read-only

STATUS current

#### DESCRIPTION

"This object represents the number of commands received by this logical unit.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of scsiLuLastCreation."

```
::= { scsiLuEntry 10 }
```

#### scsiLuReadMegaBytes OBJECT-TYPE

SYNTAX Counter32

UNITS "Megabytes"

MAX-ACCESS read-only

STATUS current

#### DESCRIPTION

"This object represents the amount of data in megabytes read from this logical unit.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of scsiLuLastCreation."

```
::= { scsiLuEntry 11 }
```

#### scsiLuWrittenMegaBytes OBJECT-TYPE

SYNTAX Counter32

UNITS "Megabytes"

MAX-ACCESS read-only

```

STATUS          current
DESCRIPTION
    "This object represents the amount of data in megabytes written
    to this logical unit.
    Discontinuities in the value of this counter can occur at re-
    initialization of the management system, and at other times as
    indicated by the value of scsiLuLastCreation."
 ::= { scsiLuEntry 12 }

scsiLuInResets OBJECT-TYPE
    SYNTAX      Counter32
    UNITS       "resets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object represents the number of times that this logical
        unit received
        - a LOGICAL UNIT RESET or TARGET RESET task management request,
        or
        - any other SCSI transport protocol-specific action or event
        that causes a Logical Unit Reset or a Hard Reset at a SCSI
        target port of the containing device
        ([SAM2] Chapters 5.9.6, 5.9.7).
        Discontinuities in the value of this counter can occur at re-
        initialization of the management system, and at other times as
        indicated by the value of scsiLuLastCreation."
    REFERENCE
        "SCSI Architecture Model-2 (SAM-2), ANSI INCITS 366-2003,
        T10 Project 1157-D, 12 September 2002 - Chapter 5.9.7 [SAM2]"
 ::= { scsiLuEntry 13 }

scsiLuOutTaskSetFullStatus OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object represents the number of Task Set full statuses
        issued for this logical unit.
        Discontinuities in the value of this counter can occur at re-
        initialization of the management system, and at other times as
        indicated by the value of scsiLuLastCreation."
 ::= { scsiLuEntry 14 }

scsiLuHSInCommands OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "commands"
    MAX-ACCESS  read-only
    STATUS      current

```

## DESCRIPTION

"This object represents the number of commands received by this logical unit. This object provides support for systems that can quickly generate a large number of commands because they run at high speed.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of scsiLuLastCreation."

```
::= { scsiLuEntry 15 }
```

## scsiLuLastCreation OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"This object indicates the value of sysUpTime when this row was last created."

```
::= { scsiLuEntry 16 }
```

## --\*\*\*\*\* Logical Unit Identifier Table \*\*\*\*\*

## scsiLuIdTable OBJECT-TYPE

SYNTAX SEQUENCE OF ScsiLuIdEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

"A table of identifiers for all logical units exposed by the local SCSI target device."

```
::= { scsiLogicalUnit 2 }
```

## scsiLuIdEntry OBJECT-TYPE

SYNTAX ScsiLuIdEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

"An entry (row) containing information applicable to a particular identifier for a particular logical unit of a particular SCSI target device within a particular SCSI instance."

INDEX {scsiInstIndex, scsiDeviceIndex, scsiLuIndex, scsiLuIdIndex}

```
::= { scsiLuIdTable 1 }
```

## ScsiLuIdEntry ::= SEQUENCE {

scsiLuIdIndex ScsiIndexValue,

scsiLuIdCodeSet ScsiIdCodeSet,

scsiLuIdAssociation ScsiIdAssociation,

scsiLuIdType ScsiIdType,

scsiLuIdValue ScsiIdValue

```
}
```

```

scsiLuIdIndex OBJECT-TYPE
    SYNTAX      ScsiIndexValue
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "This object represents an arbitrary integer used to uniquely
        identify a particular LU identifier within a particular logical
        unit within a particular SCSI target device within a particular
        SCSI instance."
 ::= { scsiLuIdEntry 1 }

scsiLuIdCodeSet OBJECT-TYPE
    SYNTAX      ScsiIdCodeSet
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "This object specifies the code set in use with this
        identifier. The value is represented in the same format as is
        contained in the identifier's Identification Descriptor within
        the logical unit's Device Identification Page."
    REFERENCE
        "ANSI - SCSI Primary Commands - 2 (SPC-2),
        ANSI INCITS 351-2001, 11 July 2001 Chapter 8: section 8.4.4,
        Vital Product Data Parameters [SPC2]"
 ::= { scsiLuIdEntry 2 }

scsiLuIdAssociation OBJECT-TYPE
    SYNTAX      ScsiIdAssociation
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "This object specifies what the identifier is associated with
        (e.g., with the addressed physical/logical device or with a
        particular port). The value is represented in the same format
        as is contained in the identifier's Identification Descriptor
        within the logical unit's Device Identification Page."
    REFERENCE
        "ANSI - SCSI Primary Commands - 2 (SPC-2),
        ANSI INCITS 351-2001, 11 July 2001, Chapter 8: section 8.4.4,
        Vital Product Data Parameters [SPC2]"
 ::= { scsiLuIdEntry 3 }

scsiLuIdType OBJECT-TYPE
    SYNTAX      ScsiIdType
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "This object specifies the type of the identifier."

```

The value is represented in the same format as is contained in the identifier's Identification Descriptor within the logical unit's Device Identification Page."

## REFERENCE

"ANSI - SCSI Primary Commands - 2 (SPC-2),  
ANSI INCITS 351-2001, 11 July 2001, Chapter 8: [section 8.4.4](#),  
Vital Product Data Parameters [[SPC2](#)]"

```
::= { scsiLuIdEntry 4 }
```

## scsiLuIdValue OBJECT-TYPE

SYNTAX ScsiIdValue

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"This object represents the actual value of this identifier. The format is defined by the objects scsiLuIdCodeSet, scsiLuIdAssociation, scsiLuIdType. The value is represented in the same format as is contained in the identifier's Identification Descriptor within the logical unit's Device Identification Page."

## REFERENCE

"ANSI - SCSI Primary Commands - 2 (SPC-2),  
ANSI INCITS 351-2001, 11 July 2001, Chapter 8: [section 8.4.4](#),  
Vital Product Data Parameters [[SPC2](#)]"

```
::= { scsiLuIdEntry 5 }
```

```
--***** The LUN Map Table *****
```

## scsiLunMapTable OBJECT-TYPE

SYNTAX SEQUENCE OF ScsiLunMapEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

"This table provides the ability to present a logical unit using different Logical Unit Numbers for different SCSI initiator devices. This table provides a mapping between a logical unit and a Logical Unit Number, and can be referenced by a ScsiAuthorizedIntrEntry to specify the LUN map for that initiator."

```
::= { scsiLogicalUnit 3 }
```

## scsiLunMapEntry OBJECT-TYPE

SYNTAX ScsiLunMapEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

"An entry containing information about the mapping of a



particular logical unit to a particular LUN. The set of entries that all have the same values of `scsiInstIndex`, `scsiDeviceIndex` and `scsiLunMapIndex` constitutes a LUN map within a particular SCSI instance.

The `StorageType` of a row in this table is specified by the instance of `scsiInstStorageType` that is INDEX-ed by the same value of `scsiInstIndex`."

INDEX { `scsiInstIndex`, `scsiDeviceIndex`, `scsiLunMapIndex`,  
`scsiLunMapLun` }

::= { `scsiLunMapTable` 1 }

```
ScsiLunMapEntry ::= SEQUENCE {
    scsiLunMapIndex      ScsiIndexValue,
    scsiLunMapLun        ScsiLUN,
    scsiLunMapLuIndex    ScsiIndexValue,
    scsiLunMapRowStatus  RowStatus
}
```

`scsiLunMapIndex` OBJECT-TYPE

SYNTAX ScsiIndexValue

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This object represents an arbitrary integer used to uniquely identify a particular `LunMap` within a particular SCSI target device within a particular SCSI instance."

::= { `scsiLunMapEntry` 1 }

`scsiLunMapLun` OBJECT-TYPE

SYNTAX ScsiLUN

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This object specifies the Logical Unit Number, to which a logical unit is mapped by this row."

::= { `scsiLunMapEntry` 2 }

`scsiLunMapLuIndex` OBJECT-TYPE

SYNTAX ScsiIndexValue

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This object identifies the logical unit for which the value of `scsiLuIndex` is the same as the value of this object. The identified logical unit is the one mapped to a LUN by this row."

::= { `scsiLunMapEntry` 3 }

```

scsiLunMapRowStatus OBJECT-TYPE
    SYNTAX          RowStatus
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        "This object allows an administrator to create and delete this
        entry."
    ::= { scsiLunMapEntry 4 }

--***** Notifications *****
-- scsiNotifications OBJECT IDENTIFIER ::= { scsiMIB 2 }

scsiNotificationsPrefix OBJECT IDENTIFIER
    ::= { scsiNotifications 0 }

scsiTgtDeviceStatusChanged NOTIFICATION-TYPE
    OBJECTS { scsiTgtDeviceStatus }
    STATUS current
    DESCRIPTION
        "This notification will be generated for each occurrence of the
        abnormal status (e.g., if the SCSI target device's current
        status is abnormal) providing that the SCSI instance's value of
        scsiInstScsiNotificationsEnable is enabled.
        An SNMP agent implementing the SCSI MIB module should not send
        more than three SCSI identical notifications in any 10-second
        period."
    ::= { scsiNotificationsPrefix 1 }

scsiLuStatusChanged NOTIFICATION-TYPE
    OBJECTS { scsiLuStatus }
    STATUS current
    DESCRIPTION
        "This notification will be generated each time that
        scsiLuStatus changes providing that the SCSI instance's value
        of scsiInstScsiNotificationsEnable is enabled.
        An SNMP agent implementing the SCSI MIB module should not send
        more than three SCSI identical notifications in any 10-second
        period."
    ::= { scsiNotificationsPrefix 2 }

--*****
-- The next part defines the conformance groups in use
-- for SCSI MIB module.
scsiCompliances OBJECT IDENTIFIER ::= { scsiConformance 1 }

scsiCompliance MODULE-COMPLIANCE

```

```
STATUS current
DESCRIPTION
    "Describes the requirements for compliance to this SCSI MIB
    module.
    If an implementation can be both a SCSI target device and a SCSI
    initiator device, all groups are mandatory."
MODULE -- this module
    MANDATORY-GROUPS {
        scsiDeviceGroup
    }

OBJECT scsiInstAlias
MIN-ACCESS read-only
DESCRIPTION
    "Write access is not mandatory."

OBJECT scsiInstScsiNotificationsEnable
MIN-ACCESS read-only
DESCRIPTION
    "Write access is not mandatory."

OBJECT scsiDeviceAlias
MIN-ACCESS read-only
DESCRIPTION
    "Write access is not mandatory."

OBJECT scsiInstStorageType
MIN-ACCESS read-only
DESCRIPTION
    "Write access is not required."

-- Conditionally mandatory groups to be included with
-- the mandatory groups when the implementation has
-- SCSI target device.

GROUP scsiTargetDeviceGroup
DESCRIPTION
    "This group is mandatory for all SCSI implementations that
    have SCSI target devices."
GROUP scsiLunMapGroup
DESCRIPTION
    "This group is mandatory for systems having the capabilities
    of mapping local SCSI target devices and logical units
    according to remote SCSI initiator devices."

OBJECT scsiAuthIntrDevOrPort
MIN-ACCESS read-only
DESCRIPTION
```

"Write access is not required."

OBJECT scsiAuthIntrName

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT scsiAuthIntrLunMapIndex

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT scsiAuthIntrRowStatus

SYNTAX RowStatus { active(1) }

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required, and only one of the six enumerated values for the RowStatus textual convention need be supported, specifically: active(1)."

GROUP scsiTgtDevLuNotificationsGroup

DESCRIPTION

"This group is mandatory for all SCSI implementations that have SCSI target devices and are able to report status changes."

-- Conditionally mandatory groups to be included with  
-- the mandatory groups when the implementation has  
-- SCSI initiator device.

GROUP scsiInitiatorDeviceGroup

DESCRIPTION

"This group is mandatory for all SCSI implementations that have SCSI initiator devices."

OBJECT scsiIntrDevTgtAccessMode

MIN-ACCESS read-only

DESCRIPTION "Write access is not mandatory."

GROUP scsiDiscoveryGroup

DESCRIPTION

"This group is mandatory for systems having the capabilities of discovering remote SCSI target devices via local SCSI initiator devices."

OBJECT scsiLunMapLuIndex

MIN-ACCESS read-only

## DESCRIPTION

"Write access is not mandatory."

OBJECT scsiLunMapRowStatus

SYNTAX RowStatus { active(1) }

MIN-ACCESS read-only

## DESCRIPTION

"Write access is not required, and only one of the six enumerated values for the RowStatus textual convention need be supported, specifically: active(1)."

OBJECT scsiDscTgtDevOrPort

MIN-ACCESS read-only

## DESCRIPTION

"Write access is not mandatory."

OBJECT scsiDscTgtName

MIN-ACCESS read-only

DESCRIPTION "Write access is not mandatory."

OBJECT scsiDscTgtConfigured

SYNTAX TruthValue { false(2) }

MIN-ACCESS read-only

## DESCRIPTION

"The value of true(1) is not mandatory neither is the write access."

OBJECT scsiDscTgtRowStatus

SYNTAX RowStatus { active(1) }

MIN-ACCESS read-only

## DESCRIPTION

"Write access is not required, and only one of the six enumerated values for the RowStatus textual convention need be supported, specifically: active(1)."

-- Conditionally mandatory groups to be included with the mandatory  
-- groups when the implementation can gather statistics.

GROUP scsiDeviceStatGroup

## DESCRIPTION

"This group is mandatory for all SCSI implementations that can gather statistics."

-- Conditionally mandatory groups to be included with the mandatory  
-- groups when the implementation can gather statistics at the SCSI  
-- initiator device side.

GROUP scsiInitiatorDevStatsGroup

## DESCRIPTION

"This group is mandatory for all SCSI implementations that can gather statistics at SCSI initiator device side."

## GROUP scsiDiscoveryStatsGroup

## DESCRIPTION

"This group is mandatory for system having the capabilities of gathering statistics regarding remote SCSI target devices via local SCSI initiator devices."

-- Conditionally mandatory groups to be included with the mandatory  
-- groups when the implementation can gather statistics at the SCSI  
-- target side.

## GROUP scsiTargetDevStatsGroup

## DESCRIPTION

"This group is mandatory for all SCSI implementations that can gather statistics at SCSI target devices."

## GROUP scsiLunMapStatsGroup

## DESCRIPTION

"This group is mandatory for SCSI implementations able to map local SCSI target devices and logical units according to remote SCSI initiator devices."

-- Conditionally mandatory groups to be included with the mandatory  
-- groups when the implementation is running at high speed and can  
-- gather statistics at the SCSI initiator device side.

## GROUP scsiInitiatorDevHSStatsGroup

## DESCRIPTION

"This group is mandatory for all SCSI implementations that can gather statistics at the SCSI initiator device side and are running at high speed, meaning speed of 4 Gbit/second or higher."

## GROUP scsiDiscoveryHSStatsGroup

## DESCRIPTION

"This group is mandatory for systems having the capabilities of gathering statistics regarding remote SCSI target devices via local SCSI initiator devices and are running at high speed, meaning speed of 4 Gbit/second or higher."

-- Conditionally mandatory groups to be included with the mandatory  
-- groups when the implementation is running at high speed and can  
-- gather statistics at the SCSI target side.

## GROUP scsiTargetDevHSStatsGroup

## DESCRIPTION

"This group is mandatory for all SCSI implementations that can gather statistics at SCSI target devices in high speed systems, meaning speed of 4 Gbit/second or higher."

GROUP scsiLunMapHSStatsGroup

DESCRIPTION

"This group is mandatory for SCSI implementations able to map local SCSI target devices and logical units according to remote SCSI initiator devices in a high speed system, meaning speed of 4 Gbit/second or higher."

::= { scsiCompliances 1 }

scsiGroups OBJECT IDENTIFIER ::= { scsiConformance 2 }

scsiDeviceGroup OBJECT-GROUP

OBJECTS {

scsiInstAlias,  
scsiInstSoftwareIndex,  
scsiInstVendorVersion,  
scsiInstScsiNotificationsEnable,  
scsiInstStorageType,  
scsiDeviceAlias,  
scsiDeviceRole,  
scsiDevicePortNumber,  
scsiPortRole,  
scsiPortTransportPtr,  
scsiTransportType,  
scsiTransportPointer,  
scsiTransportDevName

}

STATUS current

DESCRIPTION

"A collection of objects providing information about SCSI instances, devices, and ports."

::= { scsiGroups 1 }

scsiInitiatorDeviceGroup OBJECT-GROUP

OBJECTS {

scsiIntrDevTgtAccessMode,  
scsiIntrPortName,  
scsiIntrPortIdentifier,  
scsiAttTgtPortDscTgtIdx,  
scsiAttTgtPortName,  
scsiAttTgtPortIdentifier

}

STATUS current

DESCRIPTION

"This group is relevant for s SCSI initiator device and port."

```
::= { scsiGroups 2 }
```

```
scsiDiscoveryGroup OBJECT-GROUP
```

```
OBJECTS {
```

```
    scsiDscTgtDevOrPort,  
    scsiDscTgtName,  
    scsiDscTgtConfigured,  
    scsiDscTgtDiscovered,  
    scsiDscTgtRowStatus,  
    scsiDscTgtLastCreation,  
    scsiDscLunLun,  
    scsiDscLunIdCodeSet,  
    scsiDscLunIdAssociation,  
    scsiDscLunIdType,  
    scsiDscLunIdValue
```

```
}
```

```
STATUS current
```

```
DESCRIPTION
```

```
    "This group is relevant for the discovered SCSI target devices  
    by a SCSI initiator port."
```

```
::= { scsiGroups 3 }
```

```
scsiTargetDeviceGroup OBJECT-GROUP
```

```
OBJECTS {
```

```
    scsiTgtDevNumberOfLUs,  
    scsiTgtDeviceStatus,  
    scsiTgtDevNonAccessibleLUs,  
    scsiTgtPortName,  
    scsiTgtPortIdentifier,  
    scsiAttIntrPortAuthIntrIdx,  
    scsiAttIntrPortName,  
    scsiAttIntrPortIdentifier,  
    scsiLuDefaultLun,  
    scsiLuWwnName,  
    scsiLuVendorId,  
    scsiLuProductId,  
    scsiLuRevisionId,  
    scsiLuPeripheralType,  
    scsiLuStatus,  
    scsiLuState,  
    scsiLuLastCreation,  
    scsiLuIdCodeSet,  
    scsiLuIdAssociation,  
    scsiLuIdType,  
    scsiLuIdValue
```

```
}
```

```
STATUS current
```

```
DESCRIPTION
```



```
"This group is relevant for a SCSI target device and port."
 ::= { scsiGroups 4 }

scsiLunMapGroup OBJECT-GROUP
  OBJECTS {
    scsiLunMapLuIndex,
    scsiLunMapRowStatus,
    scsiAuthIntrDevOrPort,
    scsiAuthIntrName,
    scsiAuthIntrLunMapIndex,
    scsiAuthIntrLastCreation,
    scsiAuthIntrRowStatus
  }
  STATUS current
  DESCRIPTION
    "This group is a collection of attributes regarding the mapping
    between Logical Unit Number, logical unit, and target device."
 ::= { scsiGroups 5}

scsiTargetDevStatsGroup OBJECT-GROUP
  OBJECTS {
    scsiTgtDevResets,
    scsiTgtPortInCommands,
    scsiTgtPortWrittenMegaBytes,
    scsiTgtPortReadMegaBytes,
    scsiLuInCommands,
    scsiLuReadMegaBytes,
    scsiLuWrittenMegaBytes,
    scsiLuInResets,
    scsiLuOutTaskSetFullStatus
  }
  STATUS current
  DESCRIPTION
    "This group is a collection of statistics for all
    implementations of the SCSI MIB module that contain SCSI target
    devices."
 ::= { scsiGroups 6}

scsiTargetDevHSStatsGroup OBJECT-GROUP
  OBJECTS {
    scsiTgtPortHSInCommands,
    scsiLuHSInCommands
  }
  STATUS current
  DESCRIPTION
    "This group is a collection of high speed statistics for all
    implementations of the SCSI MIB module that contain SCSI target
    devices."
```

```
::= { scsiGroups 7}

scsiLunMapStatsGroup OBJECT-GROUP
    OBJECTS {
        scsiAuthIntrAttachedTimes,
        scsiAuthIntrOutCommands,
        scsiAuthIntrReadMegaBytes,
        scsiAuthIntrWrittenMegaBytes
    }
    STATUS current
    DESCRIPTION
        "This group is a collection of statistics regarding SCSI
        initiator devices authorized to attach local logical unit and
        SCSI target device."
 ::= { scsiGroups 8}

scsiLunMapHSStatsGroup OBJECT-GROUP
    OBJECTS {
        scsiAuthIntrHSOutCommands
    }
    STATUS current
    DESCRIPTION
        "This group is a collection of high speed statistics regarding
        SCSI initiator devices authorized to attach local logical unit
        and SCSI target device."
 ::= { scsiGroups 9}

scsiInitiatorDevStatsGroup OBJECT-GROUP
    OBJECTS {
        scsiIntrDevOutResets,
        scsiIntrPortOutCommands,
        scsiIntrPortWrittenMegaBytes,
        scsiIntrPortReadMegaBytes
    }
    STATUS current
    DESCRIPTION
        "This group is a collection of statistics for all
        implementations of the SCSI MIB module that contain SCSI
        initiator devices."
 ::= { scsiGroups 10}

scsiInitiatorDevHSStatsGroup OBJECT-GROUP
    OBJECTS {
        scsiIntrPortHSOutCommands
    }
    STATUS current
    DESCRIPTION
        "This group is a collection of high speed statistics for all
```

```
        implementations of the SCSI MIB module that contain SCSI
        initiator devices."
 ::= { scsiGroups 11}

scsiDiscoveryStatsGroup OBJECT-GROUP
    OBJECTS {
        scsiDscTgtInCommands,
        scsiDscTgtReadMegaBytes,
        scsiDscTgtWrittenMegaBytes
    }
    STATUS current
    DESCRIPTION
        "This group is a collection of statistics for all
        implementations of the SCSI MIB module that contain discovered
        SCSI initiator devices."
 ::= { scsiGroups 12}

scsiDiscoveryHSStatsGroup OBJECT-GROUP
    OBJECTS {
        scsiDscTgtHSInCommands
    }
    STATUS current
    DESCRIPTION
        "This group is a collection of high speed statistics for all
        implementations of the SCSI MIB module that contain discovered
        SCSI initiator devices."
 ::= { scsiGroups 13}

scsiDeviceStatGroup OBJECT-GROUP
    OBJECTS {
        scsiPortBusyStatuses
    }
    STATUS current
    DESCRIPTION
        "A collection of statistics regarding SCSI devices and
        ports."
 ::= { scsiGroups 14 }

scsiTgtDevLuNotificationsGroup NOTIFICATION-GROUP
    NOTIFICATIONS {
        scsiTgtDeviceStatusChanged,
        scsiLuStatusChanged
    }
    STATUS current
    DESCRIPTION
        "A collection of notifications regarding status change of SCSI
        target devices and logical units."
 ::= { scsiGroups 15 }
```

END

10. Object Population Example: SCSI Target and Initiator Devices on a pSCSI Bus

This section provides a sample set of values for a parallel SCSI scenario in which a SCSI MIB module can be implemented. The example shown below is not a normative part of this document and makes some assumptions about the underlying implementation, which are not based on actual implementations.

The respective sections describe the sequence of object instantiations and attempts to explain non-typical values for attributes that are unique to the scenario.

Note: While populating the objects, the population of statistics is not considered.

This scenario deals with a SCSI target and initiator devices attached to a parallel SCSI bus, defined by one of the SCSI-3 Parallel Interface standards (the version referenced in the MIB module is the 4th generation, called SPI-4). We assume that the SCSI initiator device is a Host Bus Adaptor (HBA), and the SCSI target device is a physical disk. We assume that the SCSI target device has one integrated logical unit, identified by a Logical Unit Number (LUN) of 0, which is the default LUN. The parallel SCSI transport only supports port identifiers, and not port names. The transport pointer is set to 0 since there is no MIB module defined for SPI-4.

We assume an HBA as the SCSI initiator device and a disk as the SCSI target device. We assume that the SCSI target device has one logical unit, addressed by Logical Unit Number set to 0 (LUN0), which is the default LUN. Parallel SCSI has only port identifiers, no port names. The transport pointer for parallel SCSI is set to 0 since there is no reference transport (SPI) MIB module.

Once the SCSI system is initialized, an SNMP agent should be able to view the values of variables populated in the ScsiDevice, ScsiInitiatorDevice, ScsiTargetDevice, ScsiPort, ScsiTargetPort, ScsiInitiatorPort, ScsiLogicalUnit, ScsiLUIentifier objects.

The ScsiAuthorizedIntr population depends on the transport and the implementation. As this example scenario is parallel SCSI, we deal with the ports. Hence the ScsiPortIndexOrZero is the index of the SCSI target port and ScsiAuthIntrDevOrPort is "port". Same is the case with the variables in scsiDscTgtDevOrPort.

Note that "" means zero-length string.

## 10.1. scsiInstance Table:

Attribute	Value
-----	-----
scsiInstIndex	1
scsiInstAlias	"pSCSI-1"
scsiInstSoftwareIndex	1000
scsiInstVendorVersion	"1.0a"
scsiInstScsiNotificationsEnable	true
scsiInstStorageType	nonVolatile

## 10.2. scsiDevice Table:

Attribute	Value
-----	-----
scsiInstIndex	1 1
scsiDeviceIndex	1 2
scsiDeviceAlias	"pSCSI-HBA" "pSCSI-Disk1"
scsiDeviceRole	initiator(1) target(0)
scsiDevicePortNumber	1 1

## 10.3. scsiPort Table:

Attribute	Value
-----	-----
scsiInstIndex	1 1
scsiDeviceIndex	1 2
scsiPortIndex	1 2
scsiPortRole	initiator(1) target(0)
scsiPortTransportPtr	1 2

## 10.4. scsiTransport Table:

Attribute	Value
-----	-----
scsiInstIndex	1 1
scsiDeviceIndex	1 2
scsiTransportIndex	1 2
scsiTransportType	scsiTransportSPI scsiTransportSPI
scsiTransportPointer	0.0 0.0
scsiTransportDevName	" " " "

## 10.5. scsiIntrDev Table:

Attribute	Value
-----	-----
scsiInstIndex	1
scsiDeviceIndex	1
scsiIntrDevTgtAccessMode	autoEnable(2)

## 10.6. scsiInitiatorPort Table:

Attribute	Value
-----	-----
scsiInstIndex	1
scsiDeviceIndex	1
scsiPortIndex	1
scsiIntrPortName	" "
scsiIntrPortIdentifier *1	0001b

\*1 Port Identifier for SCSI is represented by 4 bits.

## 10.7. scsiDscTgt Table:

Attribute	Value
-----	-----
scsiInstIndex	1
scsiDeviceIndex	1
scsiDscTgtIntrPortIndex	1
scsiDscTgtIndex	1
scsiDscTgtDevOrPort	port(2)
scsiDscTgtName	" "
scsiDscTgtConfigured	false(2)
scsiDscTgtDiscovered	true(1)
scsiDscTgtRowStatus	active(1)

## 10.8. scsiDscLUN:

Attribute	Value
-----	-----
scsiInstIndex	1
scsiDeviceIndex	1
scsiDscTgtIntrPortIndex	1
scsiDscTgtIndex	1
scsiDscLunIndex	1
scsiDscLunLun	0

## 10.9. scsiDscLUNIdentifier:

Attribute	Value
-----	-----
scsiInstIndex	1
scsiDeviceIndex	1
scsiDscLunIndex	1
scsiDscLunIdIndex	1
scsiDscLunIdCodeSet *1	2
scsiDscLunIdAssociation *2	1
scsiDscLunIdType *3	1
scsiDscLunIdValue	ASPENsl318203-001

\*1 - The identifier field will have ASCII graphic codes.

\*2 - The identifier is associated with the port that received the request.

\*3 - As defined in SPC. (This value specifies that the scsiDscLunIdValue contains a vendorID in the first 8 bytes concatenated with the product identifier field and product serial number.)

## 10.10. scsiAttTgtPort Table:

Attribute	Value
-----	-----
scsiInstIndex	1
scsiDeviceIndex	1
scsiPortIndex	1
scsiAttTgtPortIndex	1
scsiAttTgtPortDscTgtIdx	1
scsiAttTgtPortName	" "
scsiAttTgtPortId	0011b

## 10.11. scsiTgtDev Table:

Attribute	Value
-----	-----
scsiInstIndex	1
scsiDeviceIndex	2
scsiTgtDevNumberOfLUs	1
scsiTgtDeviceStatus	available(2)
scsiTgtDevNonAccessibleLUs	0

## 10.12. scsiTgtPort Table:

Attribute	Value
-----	-----
scsiInstIndex	1
scsiDeviceIndex	2
scsiPortIndex	2
scsiPortName	" "
scsiTgtPortIdentifier	0010b

## 10.13. scsiLU Table:

Attribute	Value
-----	-----
scsiInstIndex	1
scsiDeviceIndex	2
scsiLuIndex	1
scsiLuDefaultLun	0
scsiLuWwnName	" "
scsiLuVendorId	"xyz-corp"
scsiLuProductId	"super turbo disk"
scsiRevisionId	02
scsiLUPeripheralType	00
scsiLUStatus	available(2)
scsiLuState	exposed(3)

## 10.14. scsiLuId Table:

Attribute	Value
-----	-----
scsiInstIndex	1
scsiDeviceIndex	2
scsiLuIndex	1
scsiLuIdIndex	1
scsiLuIdCodeSet *1	2
scsiLuIdAssociation *2	1
scsiLuIdType *3	1
scsiLuIdValue	ASPENs1318203-0004

\*1 - The identifier field will have ASCII graphic codes.

\*2 - The identifier is associated with the port that received the request.

\*3 - As defined in SPC. (This value specifies that the LuIdValue contains a vendorID in the first 8 bytes concatenated with the product identifier field and product serial number.)



## 10.15. scsiLunMap Table:

Attribute	Value
-----	-----
scsiInstIndex	1
scsiDeviceIndex	2
scsiLunMapIndex	1
scsiLunMapLun	0
scsiLunMapLuIndex	1
scsiLunMapLunRowStatus	active(1)

## 10.16. scsiAuthorizedIntr Table:

Attribute	Value
-----	-----
scsiInstIndex	1
scsiDeviceIndex	2
scsiAuthIntrTgtPortIndex	2
scsiAuthIntrIndex	1
scsiAuthIntrDevOrPort	port(2)
scsiAuthIntrName	" "
scsiAuthIntrLunMapIndex	1
scsiAuthIntrRowStatus	active(1)

## 10.17. scsiAttIntrPort Table:

Attribute	Value
-----	-----
scsiInstIndex	1
scsiDeviceIndex	2
scsiPortIndex	2
scsiAttIntrPortIdx	1
scsiAttIntrPortAuthIntrIdx	1
scsiAttIntrPortName	" "
scsiAttIntrPortIdentifier	0011b

## 11. Security Considerations

There are a number of management objects defined in this MIB module that have a MAX-ACCESS clause of read-write and/or read-create. Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations. These are the following:

- o scsiInstAlias, scsiInstScsiNotificationsEnable, scsiInstStorageType and scsiDeviceAlias: these objects can be manipulated to affect the management of a SCSI instance and its

devices; specifically, the SCSI instance's administrative alias, whether it generates notifications, whether its non-default parameter settings are retained over restarts, and the administrative alias for each of its devices.

- o `scsiIntrDevTgtAccessMode`: this object can be manipulated to allow immediate access by local SCSI initiator devices to discovered SCSI target devices without waiting for administrator approval, where such approval might not be forthcoming.
- o `scsiDscTgtTable`: the objects in this table can be manipulated to remove administrator-specified controls on access by local SCSI initiator devices to discovered SCSI target devices.
- o `scsiAuthorizedIntrTable`: the objects in this table can be manipulated to remove administrator-specified controls on access by remote SCSI initiator devices to local SCSI target devices.
- o `scsiLunMapTable`: the objects in this table can be manipulated to provide access by a remote SCSI initiator device to logical units that an administrator has configured as not accessible to said initiator.

In each of the last four cases, the objects in the tables can also be manipulated to cause a denial of service attack, by preventing administrator-authorized access.

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. All seventeen of the tables in this MIB module contain information which might be considered sensitive to read access in some environments, e.g.,

- o the settings of all read-write/read-create parameter objects mentioned above,
- o `scsiInstSoftwareIndex`, `scsiInstVendorVersion`  
--which version of which software is running;
- o `scsiDeviceRole`, `scsiPortRole`, `scsiTransportType`, `scsiTransportPointer`, `scsiTransportDevName`, `scsiDscLunIdCodeSet`, `scsiDscLunIdAssociation`, `scsiDscLunIdType`, `scsiDscLunIdValue` plus information in several tables: `scsiTgtDevTable`, `scsiLuTable`, `scsiLuIdTable`, `scsiLunMapTable`

--topology information indicating which devices/ports are targets, about the transport protocols they use, and more specific information about such targets, including detailed information about the LUNs they expose and how they are mapped onto logical units;

- o scsiIntrPortOutCommands, scsiIntrPortWrittenMegaBytes, scsiIntrPortReadMegaBytes, scsiIntrPortHSOutCommands scsiDscTgtInCommands, scsiDscTgtWrittenMegaBytes, scsiDscTgtReadMegaBytes, scsiDscTgtHSInCommands, scsiTgtPortInCommands, scsiTgtPortWrittenMegaBytes, scsiTgtPortReadMegaBytes, scsiTgtPortHSInCommands, scsiAuthIntrAttachedTimes, scsiAuthIntrOutCommands, scsiAuthIntrReadMegaBytes, scsiAuthIntrWrittenMegaBytes, scsiAuthIntrHSOutCommands, scsiLuInCommands, scsiLuReadMegaBytes, scsiLuWrittenMegaBytes, scsiLuHSInCommands  
-- statistics that could be used for traffic analysis.
- o scsiAttTgtPortTable  
-- information on which initiators are connected to which targets that could be used for traffic analysis.
- o scsiAuthorizedIntrTable and scsiAttIntrPortTable tables  
-- information about which initiators are authorized to connect to that targets.

These information may need to be kept private in sensitive environments.

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.

## 12. Acknowledgements

This document is the result of the work of the SCSI MIB Group. In particular, the contributions of Sajay Selvaraj (HCL Technologies), George Penokie (IBM), and Roger Cummings (Veritas Software) were critical to the formulation of this specification.

## 13. IANA Considerations

IANA has made a MIB OID assignment under the mib-2 branch for the SCSI-MIB.

## 14. References

### 14.1. 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., and J. Schoenwaelder, "Structure of Management Information Version 2 (SMIv2)", STD 58, [RFC 2578](#), April 1999.
- [RFC2579] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Textual Conventions for SMIv2", STD 58, [RFC 2579](#), April 1999.
- [RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, [RFC 2580](#), April 1999.
- [RFC2790] Waldbusser, S. and P. Grillo, "Host Resources MIB", [RFC 2790](#), March 2000.
- [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.
- [RFC3413] Levi, D., Meyer, P., and B. Stewart, "Simple Network Management Protocol (SNMP) Applications", STD 62, [RFC 3413](#), December 2002.
- [SAM2] ANSI INCITS 366-2003, "SCSI Architecture Model-2 (SAM-2)", SAM-2 Revision 24, September 2002.
- [SPC2] ANSI INCITS 351-2001, "SCSI Primary Commands - 2 (SPC-2)", SPC-2 Revision 20, July 2001.

#### 14.2. Informative References

- [FCP2] ANSI INCITS 350-2003, "Fibre Channel Protocol for SCSI (FCP-2)", FCP-2 Revision 08, September 2002.
- [ISCSI] Bakke, M., "[Definitions of Managed Objects for iSCSI](#)", Work in Progress, October 2005.
- [RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", [RFC 3410](#), December 2002.
- [RFC3720] Satran, J., Meth, K., Sapuntzakis, C., Chadalapaka, M., and E. Zeidner, "Internet Small Computer Systems Interface (iSCSI)", [RFC 3720](#), April 2004.
- [RFC4022] Raghunarayan, R., "Management Information Base for the Transmission Control Protocol (TCP)", [RFC 4022](#), March 2005.
- [RFC4044] McCloghrie, K., "Fibre Channel Management MIB", [RFC 4044](#), May 2005.
- [SAS-1.1] T10 Project #1601-D, "Serial Attached SCSI - 1.1 (SAS-1.1)", SAS-1.1 Revision 10, September 2005.
- [SBP3] ANSI INCITS 375-2004, "Serial Bus Protocol 3 (SBP-3)", SBP-3 Revision 05, September 2003.
- [SCC2] ANSI INCITS 318-1998, "SCSI Controller Commands - 2 (SCC-2)", SCC-2 Revision 04, September 1997.
- [SPI4] ANSI INCITS 362-2002, "SCSI Parallel Interface-4 (SPI4)", SPI-4 Revision 10, May 2002.
- [SRP] ANSI INCITS 365-2002, "SCSI RDMA Protocol (SRP)", SRP Revision 16a, July 2002.

## Authors' Addresses

Michele Hallak-Stamler  
Sanrad Intelligent Storage  
27 Habarzel Street  
Tel Aviv 69710  
IL

Phone: +972 3 7674809  
EMail: [michele@sanrad.com](mailto:michele@sanrad.com)  
URI: <http://www.sanrad.com/>

Mark Bakke  
Cisco Systems, Inc.  
7900 International Drive, Suite 400  
Bloomington, MN 55425  
USA

EMail: [mbakke@cisco.com](mailto:mbakke@cisco.com)  
URI: <http://www.cisco.com/>

Yaron Lederman  
Siliquent Technologies  
21 Etzel Street  
Ramat Gan  
IL

Phone: +972 54 5308833  
EMail: [yaroned@bezeqint.net](mailto:yaroned@bezeqint.net)

Marjorie Krueger  
Hewlett-Packard  
8000 Foothills Blvd  
Roseville, CA 95747  
US

Phone: +1 916-785-2656  
EMail: [marjorie\\_krueger@hp.com](mailto:marjorie_krueger@hp.com)

Keith McCloghrie  
Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, CA 95134  
US

Phone: +1 408 526-5260  
EMail: kzm@cisco.com

## Full Copyright Statement

Copyright (C) The Internet Society (2006).

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

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

## Intellectual Property

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

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

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

## Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).