Internet Draft M. Hallak-Stamler
Sanrad Intelligent Storage
M. Bakke
Cisco Systems
K. McCloghrie
Cisco Systems
Y. Lederman
Siliquent Technologies
G. Penokie
IBM
Roger Cummings
Veritas
Sajay Selvaraj
Hcltech
Kha Sin Teow
Brocade
draft-ietf-ips-scsi-mib-00.txt January 2002

Definition of Managed Objects for SCSI Entities

Status of this Memo

Copyright Notice

Abstract

This memo defines a Management Information Base (MIB) for Small Computer System Interface (SCSI) entities, independently of the transport layer.

Table of Contents

1. The SNMP Management Framework

The SNMP Management Framework presently consists of five major components:

- o   An overall architecture, described in RFC 2571 [1].

- o Mechanisms for describing and naming objects and events for the purpose of management. The first version of this Structure of Management Information (SMI) is called SMIv1 and described in STD 16, RFC 1155 [2], STD 16, RFC 1212 [3] and RFC 1215 [4]. The second version, called SMIv2, is described in STD 58, RFC 2578 [5], STD 58, RFC 2579 [6] and STD 58, RFC 2580 [7].

- o Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in STD 15, RFC 1157 [8]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in RFC 1901 [9] and RFC 1906 [10]. The third version of the message protocol is called SNMPv3 and described in RFC 1906 [10], RFC 2572 [11] and RFC2574[12].

- o Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in STD 15, RFC 1157 [8]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [13].

- o A set of fundamental applications described in RFC 2573 [14] and the view-based access control mechanism described in RFC 2575[15].

A more detailed introduction to the current SNMP Management Framework can be found in RFC 2570 [16].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the mechanisms defined in the SMI.

This memo specifies a MIB module that is compliant to the SMIv2. A MIB conforming to the SMIv1 can be produced through the appropriate translations. The resulting translated MIB must be semantically equivalent, except where objects or events are omitted because no translation is possible (use of Counter64). Some machine readable information in SMIv2 will be converted into textual descriptions in SMIv1 during the translation process. However, this loss of machine readable information is not considered to change the semantics of the MIB.

2. Conventions

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, NOT RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in RFC 2119 [23].

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 devices, SCSI targets and Initiators and SCSI Ports. The MIB is based on documents issued by the T-10 Technical Committee and specially on SAM-2 (SCSI Architecture Model - 2) document [17].

The SCSI protocol is a client-server protocol allowing an application layer client to transmit commands to a device server and to a task manager using an interconnect subsystem.
The client side is referred as the initiator side and the server side is referred as the target side.
A target includes a collection of logical units; each logical unit has a task manager allowing an initiator to execute commands.

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 defined an 8 bit wide multi-drop “bus” structure which could interconnect a total of eight peripherals and computer systems.

It’s important to realize that all SCSI initially standardized was 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 while 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 & 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. MIBs 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, amongst other things, a definition for transporting the SCSI command sets directly over a TCP/IP infrastructure. This effort is know as iSCSI, and an iSCSI MIB is already under development.

```
                      SCSI-3 Standards Architecture (*)

                 +------------------------------+
- - - - - - - - -| Common Access Method (CAM-3) |- - - - - - - - - -
                 +------------------------------+
        +--------+    +--------+ +----------+ +--------+ +---------+
        |Reduced |    |Medium  | |Controller| |Multi-  | |Enclosure|
        |Block   |    |Changer | |Commands  | |Media   | |Services |
        |Commands|    |Commands| |(SCC,     | |Commands| | (SES)   |
        | (RBC)  |    | (SMC)  | |   SCC-2) | |(MMC,   | +---------+
        +--------+    +--------+ +----------+ | MMC-2) |      |
+--------+  |  +--------+  |           |      +--------+      |
|Block   |  |  |Stream  |  |           |           |          |
|Commands|  |  |Commands|  |           |           |          |
| (SBC)  |  |  | (SSC)  |  |           |           |          |
+--------+  |  +--------+  |           |           |          |
    |       |      |       |           |           |          |
    +---------------------------------------------------------+
                                 |
                 +-------------------------------+
                 | Primary Commands (SPC, SPC-2) |
                 +-------------------------------+
                                 |
                 +---------------------------------+
- - - - - - - - -| Architecture Model (SAM, SAM-2) |- - - - - - - -
-
                 +---------------------------------+
                                 |
       +-------------------------------------------------+
       |                      |               |          |
 +-----------+         +-------------+ +------------+    |   Parallel
 |Interlocked|         |Parallel     | |Parallel    |    |
Interface
 |Protocol   |         |Interface-2  | |Interface-3 |    |   Projects
 |   (SIP)   |         |(SPI-2)      | |(SPI-3)     |    |
 +-----------+         |             | |            |    |
       |               |[Will replace| |[New project|    |
       |               | SIP, SPI, & | | based on   |    |
       |               | Fast-20]    | | SPI-2]     |    |
 +---------+           |             | |            |    |
 |Parallel | +-------+ |             | |            |    |
 |Interface|-|Fast-20| |             | |            |    |
 |  (SPI)  | |(Ultra)| |  (Ultra2)   | |  (Ultra3)  |    |
 +---------+ +-------+ +-------------+ +------------+    |
                                                         |
                                                         |
       +-------------------------------------------------+
       |             |              |
 +----------+ +-------------+ +----------+                  Serial
 |Serial Bus| |Fibre Channel| |SSA SCSI-3|                  Interface
 |Protocol  | |Protocol     | |Protocol  |                  Projects
 | (SBP-2)  | |(FCP, FCP-2) | |(SSA-S3P) |
 +----------+ +-------------+ +----------+
       |             |              |
       |             |        +----------+
       |             |        |SSA-TL2   |
       |             |        +----------+
       |             |              |
       |             |        +----------+
 +----------+ +-------------+ |SSA-PH1 or|
 |IEEE 1394 | |Fibre Channel| |SSA-PH2   |
 +----------+ +-------------+ +----------+
```

(*) This chart reflects the currently approved SCSI-3 project family.

All projects are in T10, except Fibre Channel is in T11 and 1394 is in IEEE.

3.2 SCSI Terminology

The definitions below are part of T.10 Proposal for SAM-2 [17]. They are copied from [17].

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 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 device-resident object that connects the application client, device server or task manager to the service delivery subsystem through which requests and responses are routed. 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 target SCSI 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 device object 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 device containing logical units and SCSI target ports that receives 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 device object 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 entity residing in the target that implements a device model and processes SCSI commands sent by an application client.

3.2.9 Logical Unit Number

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

3.2.10 Nexus

A nexus is a relationship between two SCSI devices and the initiator and target objects within those SCSI devices.

```
       I_T Nexus:     A nexus between an initiator and a target
       I_T_L Nexus:   A nexus between an initiator, a target and a
                      logical unit.
       I_T_L_Q Nexus: A nexus between an initiator, a target, a
                      logical unit and a tagged task.
       I_T_L_x Nexus: Either am I_T_L nexus or an I_T_L_Q nexus.
```

3.2.11 Interconnect subsystem

One or more physical interconnects that appear as a single path for the transfer of information between SCSI devices.

3.2.12 Device Server

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

3.2.13 Task Manager

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

4.Structure of the MIB

This MIB is composed as traditionally with three main groups:

- scsiObjects
- scsiNotifications
- scsiConformance

The scsiObjects group is composed itself of five groups:

4.1 General Group

The scsiGeneral group contains the parameters general to the managed scsi entity.

4.2 Device Group

The scsiDevice group contains the characteristics of the managed SCSI device itself. Note that a SCSI entity may contain more than one SCSI device.
This group contains also the characteristics of a managed generic SCSI port. Note that a SCSI device may contain more than one SCSI port.

4.3 Initiator Group

The scsiInitiator group contains all the managed information related to an initiator device and port.

4.4 Target Group

The scsiTarget group contains all the managed information related to a target device and port.

4.5 Logical Unit Group

The scsiLogicalUnit group contains all the managed information concerning logical units, LUN hierarchy and logical unit identifiers.
This group could be located under the target group.

5.Relationship to Other MIBs

5.1 Host Resource MIB

This portion of MIB extends those managed objects to SCSI specific entities but doesn't contain reference to software like device driver. If MIB objects are required for installed packages of SCSI software, then the hrSWInstalledGroup of the host resource MIB (RFC 2790 [22]) are the standard MIB objects to use.

This memo interprets [22] as follows:

Field	Interpretation
hrSWInstalledIndex	Administrative Index.
hrSWInstalledName	See [22].
hrSWInstalledID	Pointer to the vendor MIB of the software.
hrSWInstalledType	Probably equal to deviceDriver(3)or application(4)
hrSWInstalledDate	See [22].

5.2 iSCSI MIB

To be supplied.

6. Textual Convention

6.1 Names and Identifiers

The names and the identifiers of the SCSI devices, ports and logical units depends on the underlying transport protocols; their format and length vary accordingly. Please refer to [20] in order to get more details.

Therefore, inspired by the RFC2851, textual conventions were added accordingly.

6.2 Logical Unit Number

The logical unit number is an 64-bit integer. This type does not exist in SMI and therefore, we need to define it as a textual convention for this MIB.

7. Abbreviations

This MIB will use the following abbreviations:

Inst = Instance
Dev = Device
Tgt = Target
Intr = Initiator
Att = Attached
Id = Identifier
Ident = Identifier
Idx = Index
Prt = Port
Txp = Transport

8. Warning

This paragraph will be removed in the final draft.
The following topics were not covered in the MIB yet:

  * Conformance Statement
  * Statistics

9. Object Definitions

```
SCSI-MIB DEFINITIONS ::= BEGIN

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



       scsiModule MODULE-IDENTITY
            LAST-UPDATED "200201020000Z"          -- 02 January 2002
            ORGANIZATION "IETF"
            CONTACT-INFO "
                     Michele Hallak-Stamler
                 Sanrad Intelligent Network
                 32 Habarzel Street
                 Tel Aviv, Israel

                 Phone: +972 3 7674809
                 Email: michele@sanrad.com


                 Yaron Lederman
                 Siliquent Technologies Ltd.
                 33 Bezalel Street
                 Ramat Gan, Israel

                 Phone: +972 3 7552320
                 Email: yaronl@siliquent.com

                    "
            DESCRIPTION  "The SCSI MIB"
       -- Revision History

            REVISION     "200201020000Z"
       DESCRIPTION  " First Draft. Reflects the object model only
       and doesn't include statistics yet."
       ::= { experimental xxx }   -- must be changed in the future


  -- Textual Conventions
  ScsiLUNFormat  ::= TEXTUAL-CONVENTION
       STATUS current
       DESCRIPTION
        "It is:
            -  a zero-length octet string or
            -  a two-bytes octets if the underlying transport
               protocol is SBP-3 or SPI-4 using data group transfers
               or
            -  an eight bytes integer for all other cases."
       SYNTAX OCTET STRING (SIZE (0 | 2 | 8))

  ScsiIndexValue ::= TEXTUAL-CONVENTION
       STATUS current
       DESCRIPTION
       "A number greater than zero for administrative indices in a
       table."
       SYNTAX    Unsigned32(1..4294967295)

  ScsiPortIndexValueOrZero ::= TEXTUAL-CONVENTION
       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.  The value zero is object-specific and
       must therefore be defined as part of the description of any
       object, which 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
         "Denotes a generic SCSI device or port identifier.
       The format depends on the transport used:
       - SPI: only bits:0-3   for a port identifier
       - SPI: identifier of a device is a null-length octet string.
       - FCP: 3 bytes for a port identifier
       - FCP: identifier of a device is a null-length octet string.
       - SRP: 16 bytes identifier for a port.
       - SRP: identifier of a device is a null-length octet string.
       - iSCSI: 256 bytes for a device identifier.
       - iSCSI: 258 bytes for a target port.
       - iSCSI: 262 bytes for an initiator port.
       - SBP: identifier of a device is a null-length octet string.
       - SBP: 2 bytes for an initiator port identifier.
       - SBP: 11 bytes for a target port identifier. "
  SYNTAX OCTET STRING (SIZE (0 | 1 | 2 | 3| 11 | 16 | 256| 258|262))

  ScsiName ::= TEXTUAL-CONVENTION
            STATUS       current
            DESCRIPTION
         "Denotes a generic SCSI device or port name.
         The format depends on the transport used:
       - SPI: name of a device or a port is a null-length octet
       string.
       - FCP: 8 bytes for a port name.
       - FCP: name of a device is a null-length octet string.
       - SRP: 16 bytes name for a port.
       - SRP: name of a device is a null-length octet string.
       - iSCSI: 256 bytes for a device name.
       - iSCSI: 258 bytes for a target port.
       - iSCSI: 262 bytes for an initiator port.
       - SBP: name of a device is a null-length octet string.
       - SBP: 8 bytes for an initiator port name.
       - SBP: 11 bytes for a target port name. "

   SYNTAX OCTET STRING (SIZE (0 | 8 | 11 |16 | 256 | 258| 262))

  ScsiDeviceOrPort ::= TEXTUAL-CONVENTION
            STATUS    current
            DESCRIPTION
       "This type allows to decide if some configuration is
       applicable to a port or to a device."
       SYNTAX         INTEGER   {
                      device(1),
                      port(2),
                      other(3)
                  }

  scsiObjects    OBJECT IDENTIFIER ::= { scsiModule 1 }
  scsiNotifications OBJECT IDENTIFIER ::= { scsiModule 2 }
  scsiConformance     OBJECT IDENTIFIER ::= { scsiModule 3 }

  scsiTransportTypes  OBJECT IDENTIFIER ::= { scsiObjects 1 }
  scsiGeneral    OBJECT IDENTIFIER ::= { scsiObjects 2 }
  scsiDevice          OBJECT IDENTIFIER ::= { scsiObjects 3 }
  scsiInitiator  OBJECT IDENTIFIER ::= { scsiObjects 4 }
  scsiTarget          OBJECT IDENTIFIER ::= { scsiObjects 5 }
       scsiLogicalUnit     OBJECT IDENTIFIER ::= { scsiTarget 8 }


  -- Transport Types
  scsiTranportOther   OBJECT IDENTIFIER ::= { scsiTransportTypes 1 }
  scsiTranportSPI     OBJECT IDENTIFIER ::= { scsiTransportTypes 2 }
  scsiTransportFCP    OBJECT IDENTIFIER ::= { scsiTransportTypes 3 }
  scsiTransportSRP    OBJECT IDENTIFIER ::= { scsiTransportTypes 4 }
  scsiTransportISCSI  OBJECT IDENTIFIER ::= { scsiTransportTypes 5 }
  scsiTransportSBP    OBJECT IDENTIFIER ::= { scsiTransportTypes 6 }

  -- Comparatively to iSCSI MIB, I'm removing one level of OBJECT ID
  -- tree.
  scsiGenInstanceTable OBJECT-TYPE
       SYNTAX           SEQUENCE OF ScsiGenInstanceEntry
       MAX-ACCESS       not-accessible
       STATUS           current
       DESCRIPTION
        "A list of SCSI instances present on the system."
  ::= { scsiGeneral 1 }

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

  ScsiGenInstanceEntry::= SEQUENCE {
       scsiInstIndex                 ScsiIndexValue,
       scsiInstAlias                 SnmpAdminString,
       scsiInstReference             Integer32,
       scsiInstVendorVersion         SnmpAdminString,
       scsiInstScsiDeviceNumber      Unsigned32
    }

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

  scsiInstAlias OBJECT-TYPE
       SYNTAX         SnmpAdminString
       MAX-ACCESS     read-write
       STATUS         current
       DESCRIPTION
       "An administrative string, configured by the administrator to
       the usage of the administrator. Can be a zero-length string."
  ::= { scsiGenInstanceEntry 2 }

  scsiInstReference    OBJECT-TYPE
       SYNTAX         Integer32
       MAX-ACCESS     read-only
       STATUS         current
       DESCRIPTION
       "The index in the hrSWInstalledTable of RFC 2790
       corresponding to this software entity."
  ::= { scsiGenInstanceEntry 3 }

  scsiInstVendorVersion OBJECT-TYPE
       SYNTAX         SnmpAdminString
       MAX-ACCESS     read-only
       STATUS         current
       DESCRIPTION
       "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."
  ::= { scsiGenInstanceEntry 4 }

  scsiInstScsiDeviceNumber OBJECT-TYPE
       SYNTAX         Unsigned32 (1..4294967295)
       MAX-ACCESS     read-only
       STATUS         current
       DESCRIPTION
        "The number of SCSI Device currently associated with this
       SCSI instance."
  ::= { scsiGenInstanceEntry 5 }
```

```
-- Corresponding Statistics
--     scsiGenInstStatTable OBJECT-TYPE
--     SYNTAX           SEQUENCE OF ScsiGenInstStatEntry
--     MAX-ACCESS       not-accessible
--     STATUS           current
--     DESCRIPTION
--        "A list of statistics referring to SCSI instances present
--     on the system."
--     ::= { scsiGeneral 2 }


-- SCSI Devices
scsiDeviceTable  OBJECT-TYPE
       SYNTAX           SEQUENCE OF ScsiDeviceEntry
       MAX-ACCESS       not-accessible
       STATUS           current
       DESCRIPTION
        "A list of SCSI Devices present on the system."
  ::= { scsiDevice 1 }

  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."
       INDEX { scsiInstIndex, scsiDeviceIndex }
  ::= { scsiDeviceTable 1 }

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

  scsiDeviceIndex OBJECT-TYPE
       SYNTAX         ScsiIndexValue
       MAX-ACCESS     not-accessible
       STATUS         current
       DESCRIPTION
       "An arbitrary index for this device."
  ::= { scsiDeviceEntry 1 }

  scsiDeviceAlias OBJECT-TYPE
       SYNTAX         SnmpAdminString
       MAX-ACCESS     read-write
       STATUS         current
       DESCRIPTION
       "An admistrative name for this device. May be empty."
  ::= { scsiDeviceEntry 2 }

  scsiDeviceRole OBJECT-TYPE
       SYNTAX         BITS {
                      target(0),
                      initiator(1)
                      }
       MAX-ACCESS     read-only
       STATUS         current
       DESCRIPTION
       "Is this device acting as an initiator, or as a target or as
     both."
  ::= { scsiDeviceEntry 3 }

  scsiDeviceSerialNumber OBJECT-TYPE
       SYNTAX         SnmpAdminString
       MAX-ACCESS     read-only
       STATUS         current
       DESCRIPTION
       " The serial number of this SCSI device."
  ::= { scsiDeviceEntry 4}

  scsiDevicePortNumber OBJECT-TYPE
       SYNTAX         Unsigned32
       MAX-ACCESS     read-only
       STATUS         current
       DESCRIPTION
       "The number of ports contained in this device."
  ::= { scsiDeviceEntry 5 }


  -- Ports 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 the instance."
  ::= { scsiDevice 2 }

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

  scsiPortIndex OBJECT-TYPE
       SYNTAX         ScsiIndexValue
       MAX-ACCESS     not-accessible
       STATUS         current
       DESCRIPTION
       "An arbitrary index for this port."
  ::= { scsiPortEntry 1 }

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


  -- Management Objects regarding initiators
  scsiInitiatorDeviceTable  OBJECT-TYPE
       SYNTAX           SEQUENCE OF ScsiInitiatorDeviceEntry
       MAX-ACCESS       not-accessible
       STATUS           current
       DESCRIPTION
        "This table contains the parameters for each initiator
       device."
   ::= { scsiInitiator 1}

  scsiInitiatorDeviceEntry OBJECT-TYPE
       SYNTAX              ScsiInitiatorDeviceEntry
       MAX-ACCESS          not-accessible
       STATUS              current
       DESCRIPTION
       "An entry (row) containing parameters applicable to an
       initiator device."
       INDEX { scsiInstIndex, scsiDeviceIndex }
  ::= { scsiInitiatorDeviceTable  1 }

  ScsiInitiatorDeviceEntry ::= SEQUENCE {
       scsiInitiatorDeviceResetSent      Counter32
       -- More statistics to be placed here
  }

  scsiInitiatorDeviceResetSent OBJECT-TYPE
```

```
       SYNTAX         Counter32
       MAX-ACCESS     not-accessible
       STATUS         current
       DESCRIPTION
       "Number of reset sent by this initiator device since its
       own last reset."
       ::= { scsiInitiatorDeviceEntry 1 }

  scsiIntrTxportTable OBJECT-TYPE
       SYNTAX           SEQUENCE OF ScsiIntrTxportEntry
       MAX-ACCESS       not-accessible
       STATUS           current
       DESCRIPTION
        "This table contains a list of transports in use with this
        initiator device."
  ::= { scsiInitiator 2 }

  scsiIntrTxportEntry OBJECT-TYPE
       SYNTAX              ScsiIntrTxportEntry
       MAX-ACCESS          not-accessible
       STATUS              current
       DESCRIPTION
       "An entry (row) containing parameters applicable to a
       transport used by this initiator device.”
       INDEX { scsiInstIndex, scsiDeviceIndex, scsiIntrTxportIndex}
  ::= { scsiIntrTxportTable 1 }

  ScsiIntrTxportEntry ::= SEQUENCE {
       scsiIntrTxportIndex      ScsiIndexValue,
       scsiIntrTxportPointer    RowPointer,
       scsiIntrTxportType       AutonomousType,
       scsiIntrTxportDevName    ScsiName,
       scsiIntrTxportDevId      ScsiIdentifier
    }

  scsiIntrTxportIndex      OBJECT-TYPE
       SYNTAX         ScsiIndexValue
       MAX-ACCESS     not-accessible
       STATUS         current
       DESCRIPTION
       "An administrative index for this transport."
       ::= { scsiIntrTxportEntry 1 }

  scsiIntrTxportPointer    OBJECT-TYPE
       SYNTAX         RowPointer
       MAX-ACCESS     read-only
       STATUS         current
       DESCRIPTION
       "A pointer to an instance in a MIB of the transport
       corresponding to this entry.
       For example, if the transport of this device is iSCSI, this
       field will point to the corresponding iSCSI Managed Object.
       If there is no MIB for this transport, the null OID 0.0
       should be returned."
       ::= { scsiIntrTxportEntry 2 }

  scsiIntrTxportType  OBJECT-TYPE
       SYNTAX         AutonomousType
       MAX-ACCESS     read-only
       STATUS         current
       DESCRIPTION
       "The type of ScsiName for this device when using this
       transport."
       ::= { scsiIntrTxportEntry 3 }

  scsiIntrTxportDevName    OBJECT-TYPE
       SYNTAX         ScsiName
       MAX-ACCESS     read-only
       STATUS         current
       DESCRIPTION
       "The name of this device according the type of transport."
       ::= { scsiIntrTxportEntry 4 }

  scsiIntrTxportDevId OBJECT-TYPE
       SYNTAX         ScsiIdentifier
       MAX-ACCESS     read-only
       STATUS         current
       DESCRIPTION
       "The identifier of this device according the type of
       transport."
       ::= { scsiIntrTxportEntry 5 }


  -- The following section describes managed objects related to
  -- initiator ports.
  scsiInitiatorPortTable OBJECT-TYPE
       SYNTAX           SEQUENCE OF ScsiInitiatorPortEntry
       MAX-ACCESS       not-accessible
       STATUS           current
       DESCRIPTION
        "This table contains all the initiator ports of each SCSI
       Initiator or Target/Initiator device."
  ::= { scsiInitiator 3 }

  scsiInitiatorPortEntry OBJECT-TYPE
       SYNTAX              ScsiInitiatorPortEntry
       MAX-ACCESS          not-accessible
       STATUS              current
       DESCRIPTION
       "An entry (row) containing parameters applicable to the
       corresponding initiator port. "
     INDEX { scsiInstIndex, scsiDeviceIndex, scsiPortIndex }
  ::= { scsiInitiatorPortTable 1 }
  ScsiInitiatorPortEntry ::= SEQUENCE {
       scsiInitiatorPortTxportType        AutonomousType,
       scsiInitiatorPortName              ScsiName,
       scsiInitiatorPortIdentifier        ScsiIdentifier,
       scsiInitiatorPortTxportPointer     RowPointer
    }

  scsiInitiatorPortTxportType OBJECT-TYPE
       SYNTAX              AutonomousType
       MAX-ACCESS          read-only
       STATUS              current
       DESCRIPTION
       "The type of name for the port according transport protocol."
       ::= { scsiInitiatorPortEntry 1 }

  scsiInitiatorPortName OBJECT-TYPE
       SYNTAX              ScsiName
       MAX-ACCESS          read-only
       STATUS              current
       DESCRIPTION
       "The name of the port assigned by the SCSI protocol."
       ::= { scsiInitiatorPortEntry 2 }


  scsiInitiatorPortIdentifier OBJECT-TYPE
       SYNTAX              ScsiIdentifier
       MAX-ACCESS          read-only
       STATUS              current
       DESCRIPTION
       "The identifier of the port according the type of transport."
       ::= { scsiInitiatorPortEntry 3 }

  scsiInitiatorPortTxportPointer OBJECT-TYPE
       SYNTAX              RowPointer
       MAX-ACCESS          read-only
       STATUS              current
       DESCRIPTION
       " This shall be the Object Identifier of the corresponding
       port definition in the correct transport MIB, or if that
       information is not accessible, this shall be the base Object
       Identifier of the transport MIB for this port's transport (in
       order to identify the transport type of this port).  For
       instance, if the transport of this port is iSCSI, this shall
       be the OID of this port in the iSCSI MIB, or the base OID for
       the iSCSI transport MIB."
       ::= { scsiInitiatorPortEntry 4 }


  -- Statistics for Initiator ports will be placed here.
  -- scsiIntrPortStatTable OBJECT-TYPE
  --   SYNTAX             SEQUENCE OF ScsiIntrPortStatEntry
  --   MAX-ACCESS         not-accessible
  --   STATUS           current
  --   DESCRIPTION
  --   "This table contains statistics for all the initiator ports -
  --   of each SCSI Initiator or Target/Initiator device."
  --   ::= { scsiInitiator 4}

  -- Attached Targets to Initiator Port Table
  scsiIntrAttTgtPortTable OBJECT-TYPE
       SYNTAX           SEQUENCE OF ScsiIntrAttTgtPortEntry
       MAX-ACCESS       not-accessible
       STATUS           current
       DESCRIPTION
        "This table includes all the remote (not in the current
       system) target ports that are currently attached to each
       local initiator port of this entity."
  ::= { scsiInitiator 5 }

  scsiIntrAttTgtPortEntry OBJECT-TYPE
       SYNTAX              ScsiIntrAttTgtPortEntry
       MAX-ACCESS          not-accessible
       STATUS              current
       DESCRIPTION
       "An entry (row)represents a remote target port currently
       attached to the local initiator port corresponding to the
       scsiPortIndex. "
       INDEX { scsiInstIndex, scsiDeviceIndex, scsiPortIndex,
       scsiIntrAttTgtPortIndex }
  ::= { scsiIntrAttTgtPortTable 1 }

  ScsiIntrAttTgtPortEntry ::= SEQUENCE {
       scsiIntrAttTgtPortIndex       ScsiIndexValue,
       scsiIntrAttTgtPortName        ScsiName,
       scsiIntrAttTgtPortIdentifier  ScsiIdentifier
    }

  scsiIntrAttTgtPortIndex  OBJECT-TYPE
       SYNTAX              ScsiIndexValue
       MAX-ACCESS          not-accessible
       STATUS              current
       DESCRIPTION
       "An administrative index for this current attached target."
  ::= { scsiIntrAttTgtPortEntry 1 }

  scsiIntrAttTgtPortName OBJECT-TYPE
       SYNTAX              ScsiName
       MAX-ACCESS          read-only
       STATUS              current
       DESCRIPTION
       "The name of the remote target attached to this initiator
       port."
  ::= { scsiIntrAttTgtPortEntry 2 }
```

```
   scsiIntrAttTgtPortIdentifier OBJECT-TYPE
        SYNTAX              ScsiIdentifier
        MAX-ACCESS          read-only
        STATUS              current
        DESCRIPTION
        "The identifier of the remote target attached to this local
        initiator port according the type of transport."
   ::= { scsiIntrAttTgtPortEntry 3 }

   -- Statistics per target attached port to local initiator port
   -- scsiIntrAttTgtPrtStatTable OBJECT-TYPE
   --   SYNTAX           SEQUENCE OF ScsiIntrAttTgtPrtStatEntry
   --   MAX-ACCESS       not-accessible
   --   STATUS           current
   --   DESCRIPTION
   --   "This table includes statistics for all the remote (not in
   --   the current system) target ports that are currently attached
   --   to each local initiator port of this entity."
   --   ::= { scsiInitiator 6 }

   -- Management Objects regarding target type of scsi devices
   --
   scsiTargetDeviceTable OBJECT-TYPE
        SYNTAX           SEQUENCE OF ScsiTargetDeviceEntry
        MAX-ACCESS       not-accessible
        STATUS           current
        DESCRIPTION
         "This table is an extension of the device table including
        parameters specific to a target."
   ::= { scsiTarget 1 }

   scsiTargetDeviceEntry OBJECT-TYPE
        SYNTAX              ScsiTargetDeviceEntry
        MAX-ACCESS          not-accessible
        STATUS              current
        DESCRIPTION
        "An entry (row) containing parameters applicable to the
        corresponding target device. "
        INDEX { scsiInstIndex, scsiDeviceIndex }
   ::= { scsiTargetDeviceTable 1 }

   ScsiTargetDeviceEntry ::= SEQUENCE {
        scsiTargetDeviceNumberOfLogicalUnits    Unsigned32
     }

   scsiTargetDeviceNumberOfLogicalUnits  OBJECT-TYPE
        SYNTAX              Unsigned32
        MAX-ACCESS          read-only
        STATUS              current
        DESCRIPTION
        "Number of Logical Units associated with this target."
```

```
  ::= { scsiTargetDeviceEntry 1 }

-- Statistics per target device will be placed here
--          scsiTargetDevStatTable OBJECT-TYPE
--          SYNTAX           SEQUENCE OF ScsiTargetDevStatEntry
--          MAX-ACCESS       not-accessible
--          STATUS           current
--          DESCRIPTION
--          "This table is an extension of the device table
--          including statistics specific to a target device."
--          ::= { scsiTarget 2 }

  scsiTargetTxportTable OBJECT-TYPE
       SYNTAX           SEQUENCE OF ScsiTargetTxportEntry
       MAX-ACCESS       not-accessible
       STATUS           current
       DESCRIPTION
        "This table contains a list of transports in use with this
        target device."
  ::= { scsiTarget 3 }

  scsiTargetTxportEntry OBJECT-TYPE
       SYNTAX              ScsiTargetTxportEntry
       MAX-ACCESS          not-accessible
       STATUS              current
       DESCRIPTION
       "An entry (row) containing parameters applicable to a
       transport used by this target device."
       INDEX { scsiInstIndex, scsiDeviceIndex, scsiTargetTxportIndex
     }
  ::= { scsiTargetTxportTable 1 }

  ScsiTargetTxportEntry ::= SEQUENCE {
       scsiTargetTxportIndex        ScsiIndexValue,
       scsiTargetTxportPointer      RowPointer,
       scsiTargetTxportType         AutonomousType,
       scsiTargetTxportDevName      ScsiName,
       scsiTargetTxportDevId        ScsiIdentifier
    }

  scsiTargetTxportIndex OBJECT-TYPE
       SYNTAX         ScsiIndexValue
       MAX-ACCESS     not-accessible
       STATUS         current
       DESCRIPTION
       "An administrative index for this transport."
       ::= { scsiTargetTxportEntry 1 }

  scsiTargetTxportPointer OBJECT-TYPE
       SYNTAX         RowPointer
       MAX-ACCESS     read-only
       STATUS         current
       DESCRIPTION
       "A pointer to an instance in a MIB of the transport
       corresponding to this entry.
       For example, if the transport of this device is iSCSI, this
       field will point to the corresponding iSCSI Managed Object.
       If there is no MIB for this transport, the null OID 0.0
       should be returned."
       ::= { scsiTargetTxportEntry 2 }

  scsiTargetTxportType OBJECT-TYPE
       SYNTAX         AutonomousType
       MAX-ACCESS     read-only
       STATUS         current
       DESCRIPTION
       "The type of name used by this transport."
       ::= { scsiTargetTxportEntry 3 }

  scsiTargetTxportDevName OBJECT-TYPE
       SYNTAX         ScsiName
       MAX-ACCESS     read-only
       STATUS         current
       DESCRIPTION
       "The name of this device according its transport type."
       ::= { scsiTargetTxportEntry 4 }

  scsiTargetTxportDevId    OBJECT-TYPE
       SYNTAX         ScsiIdentifier
       MAX-ACCESS     read-only
       STATUS         current
       DESCRIPTION
       "The identifier of this device according its transport."
       ::= { scsiTargetTxportEntry 5 }

  -- Target Port Table
  scsiTargetPortTable OBJECT-TYPE
       SYNTAX           SEQUENCE OF ScsiTargetPortEntry
       MAX-ACCESS       not-accessible
       STATUS           current
       DESCRIPTION
        "This table includes all the target ports of all the SCSI
       target devices."
  ::= { scsiTarget 4 }

  scsiTargetPortEntry OBJECT-TYPE
       SYNTAX              ScsiTargetPortEntry
       MAX-ACCESS          not-accessible
       STATUS              current
       DESCRIPTION
       "An entry (row) containing parameters applicable to the
       corresponding target device. "
       INDEX { scsiInstIndex, scsiDeviceIndex, scsiPortIndex}
  ::= { scsiTargetPortTable 1 }
  ScsiTargetPortEntry ::= SEQUENCE {
       scsiTargetPortTxportType      AutonomousType,
       scsiTargetPortName            ScsiName,
       scsiTargetPortIdentifier      ScsiIdentifier,
       scsiTargetPortTxportPointer        RowPointer
    }

  scsiTargetPortTxportType OBJECT-TYPE
       SYNTAX              AutonomousType
       MAX-ACCESS          read-only
       STATUS              current
       DESCRIPTION
       "The type of name for the port according protocol."
       ::= { scsiTargetPortEntry 1 }

  scsiTargetPortName OBJECT-TYPE
       SYNTAX              ScsiName
       MAX-ACCESS          read-only
       STATUS              current
       DESCRIPTION
       "The name of the port assigned by the SCSI protocol."
       ::= { scsiTargetPortEntry 2 }

  scsiTargetPortIdentifier OBJECT-TYPE
       SYNTAX              ScsiIdentifier
       MAX-ACCESS          read-only
       STATUS              current
       DESCRIPTION
       "The identifier of the port according the type of transport."
       ::= { scsiTargetPortEntry 3 }

  scsiTargetPortTxportPointer OBJECT-TYPE
       SYNTAX              RowPointer
       MAX-ACCESS          read-only
       STATUS              current
       DESCRIPTION
       " This shall be the Object Identifier of the corresponding
       port definition in the correct transport MIB, or if that
       information is not accessible, this shall be the base Object
       Identifier of the transport MIB for this port's transport (in
       order to identify the transport type of this port).  For
       instance, if the transport of this port is iSCSI, this shall
       be the OID of this port in the iSCSI MIB, or the base OID for
       the iSCSI transport MIB."
       ::= { scsiTargetPortEntry 4 }


  -- Target Port Statistic Table will be placed here
  --   scsiTargetPortStatTable OBJECT-TYPE
  --   SYNTAX           SEQUENCE OF ScsiTargetPortStatEntry
  --   MAX-ACCESS       not-accessible
  --   STATUS           current
  --   DESCRIPTION
  --   "This table includes the statistics for the target ports of
  --   all the SCSI target devices."
  --   ::= { scsiTarget 5 }

  -- The following table should contain those initiator ports that
  -- may be attached to specific target ports and on which, an
  -- administrator would like to keep permanent information and long
  -- term statistic even when not currently attached.
  scsiTgtPrtIntrPrtTable OBJECT-TYPE
       SYNTAX           SEQUENCE OF ScsiTgtPrtIntrPrtEntry
       MAX-ACCESS       not-accessible
       STATUS           current
       DESCRIPTION
        "This table includes all the configured initiator ports that
       may attach a target port of this entity and that may interest
       an administrator."
  ::= { scsiTarget 6 }

  scsiTgtPrtIntrPrtEntry OBJECT-TYPE
       SYNTAX              ScsiTgtPrtIntrPrtEntry
       MAX-ACCESS          not-accessible
       STATUS              current
       DESCRIPTION
       "An entry (row)represents an initiator port that may attach
       the target port corresponding to the scsiPortIndex. "
       INDEX { scsiInstIndex, scsiDeviceIndex, scsiPortIndex,
       scsiTgtPrtIntrPrtIndex }
  ::= { scsiTgtPrtIntrPrtTable 1 }

  ScsiTgtPrtIntrPrtEntry ::= SEQUENCE {
       scsiTgtPrtIntrPrtIndex        ScsiIndexValue,
       scsiTgtPrtIntrPrtPortName     ScsiName,
       scsiTgtPrtIntrPrtRowStatus    RowStatus
    }

  scsiTgtPrtIntrPrtIndex   OBJECT-TYPE
       SYNTAX              ScsiIndexValue
       MAX-ACCESS          not-accessible
       STATUS              current
       DESCRIPTION
       "An administrative index for this configured initiator port."
       ::= { scsiTgtPrtIntrPrtEntry 1 }

  scsiTgtPrtIntrPrtPortName OBJECT-TYPE
       SYNTAX              ScsiName
       MAX-ACCESS          read-create
       STATUS              current
       DESCRIPTION
       "The name of the initiator port configured to this target
       port."
       ::= { scsiTgtPrtIntrPrtEntry 2 }

  scsiTgtPrtIntrPrtRowStatus OBJECT-TYPE
       SYNTAX              RowStatus
       MAX-ACCESS          read-create
       STATUS              current
       DESCRIPTION
       "This field allows an administrator to create or delete this
       entry."
       ::= { scsiTgtPrtIntrPrtEntry 3 }

  -- Attached Initiators to Target Table
  scsiTgtAttIntrPortTable OBJECT-TYPE
       SYNTAX           SEQUENCE OF ScsiTgtAttIntrPortEntry
       MAX-ACCESS       not-accessible
       STATUS           current
       DESCRIPTION
        "This table includes all the remote initiator ports that are
       currently attached to each local target port of this local
       entity."
  ::= { scsiTarget 7 }

  scsiTgtAttIntrPortEntry OBJECT-TYPE
       SYNTAX              ScsiTgtAttIntrPortEntry
       MAX-ACCESS          not-accessible
       STATUS              current
       DESCRIPTION
       "An entry (row)represents a remote initiator currently
       attached to the local target port corresponding to the
       scsiPortIndex. "
       INDEX { scsiInstIndex, scsiDeviceIndex, scsiPortIndex,
       scsiTgtAttIntrIndex }
  ::= { scsiTgtAttIntrPortTable 1 }

  ScsiTgtAttIntrPortEntry ::= SEQUENCE {
       scsiTgtAttIntrIndex                ScsiIndexValue,
       scsiTgtAttIntrTgtPrtIntrPrtIdx     ScsiPortIndexValueOrZero,
       scsiTgtAttIntrPortName             ScsiName,
       scsiTgtAttIntrPortId               ScsiIdentifier
    }

  scsiTgtAttIntrIndex OBJECT-TYPE
       SYNTAX              ScsiIndexValue
       MAX-ACCESS          not-accessible
       STATUS              current
       DESCRIPTION
       "An administrative index for this current attached
       initiator."
       ::= { scsiTgtAttIntrPortEntry 1 }

  scsiTgtAttIntrTgtPrtIntrPrtIdx OBJECT-TYPE
       SYNTAX              ScsiPortIndexValueOrZero
       MAX-ACCESS          read-only
       STATUS              current
       DESCRIPTION
       "This field is the index of the configured entry in the
       scsiTgtPrtIntrPrtTable corresponding to this remote attached
       initiator port. If it doesn't exist, this field should return
       0."
       ::= { scsiTgtAttIntrPortEntry 2 }

  scsiTgtAttIntrPortName OBJECT-TYPE
       SYNTAX              ScsiName
       MAX-ACCESS          read-only
       STATUS              current
       DESCRIPTION
       "The name of the remote initiator attached to this local
       target port."
       ::= { scsiTgtAttIntrPortEntry 3 }

  scsiTgtAttIntrPortId OBJECT-TYPE
       SYNTAX              ScsiIdentifier
       MAX-ACCESS          read-only
       STATUS              current
       DESCRIPTION
       "The identifier of the remote initiator attached to this
       local target port."
       ::= { scsiTgtAttIntrPortEntry 4 }


  -- Managed Objects regarding logical units
  scsiLogicalUnitTable OBJECT-TYPE
       SYNTAX           SEQUENCE OF ScsiLogicalUnitEntry
       MAX-ACCESS       not-accessible
       STATUS           current
       DESCRIPTION
        "This table includes all the logical units exposed by a
       target device."
  ::= { scsiLogicalUnit 1 }

  scsiLogicalUnitEntry OBJECT-TYPE
       SYNTAX              ScsiLogicalUnitEntry
       MAX-ACCESS          not-accessible
       STATUS              current
       DESCRIPTION
       "An entry (row) containing parameters applicable to the
       corresponding logical unit. "
       INDEX { scsiInstIndex, scsiDeviceIndex, scsiLUIndex}
  ::= { scsiLogicalUnitTable 1 }

  ScsiLogicalUnitEntry ::= SEQUENCE {
       scsiLUIndex         ScsiIndexValue,
       scsiLUDefaultLun    ScsiLUNFormat,
       scsiLUName          ScsiLUNFormat,
       scsiLUVendorId      SnmpAdminString,
       scsiLUProductId     SnmpAdminString,
       scsiLURevisionId    SnmpAdminString
    }

  scsiLUIndex         OBJECT-TYPE
       SYNTAX              ScsiIndexValue
       MAX-ACCESS          not-accessible
       STATUS              current
       DESCRIPTION
       "Administrative index in the Logical Unit table."
  ::= { scsiLogicalUnitEntry 1 }

  scsiLUDefaultLun         OBJECT-TYPE
       SYNTAX              ScsiLUNFormat
       MAX-ACCESS          read-only
       STATUS              current
       DESCRIPTION
       "The default Logical Unit Number for this Logical Unit."
       ::= { scsiLogicalUnitEntry 2 }

  scsiLUName          OBJECT-TYPE
       SYNTAX              ScsiLUNFormat
       MAX-ACCESS          read-only
       STATUS              current
       DESCRIPTION
       "The World-Wide Name of this LU."
       ::= { scsiLogicalUnitEntry 3 }

  scsiLUVendorId      OBJECT-TYPE
       SYNTAX              SnmpAdminString
       MAX-ACCESS          read-only
       STATUS              current
       DESCRIPTION
       "A string identifying the vendor of this LU according to the
       value in SCSI device page."
       ::= { scsiLogicalUnitEntry 4 }

  scsiLUProductId     OBJECT-TYPE
       SYNTAX              SnmpAdminString
       MAX-ACCESS          read-only
       STATUS              current
       DESCRIPTION
       "A string identifying the product for this LU according to
       the value in SCSI device page."
       ::= { scsiLogicalUnitEntry 5 }

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

```
       "A string defining the product revision of this LU according
       to the value in SCSI device page."
            ::= { scsiLogicalUnitEntry 6 }


  scsiLUIdentifierTable OBJECT-TYPE
       SYNTAX           SEQUENCE OF ScsiLUIdentifierEntry
       MAX-ACCESS       not-accessible
       STATUS           current
       DESCRIPTION
        "A table of identifiers per logical unit."
  ::= { scsiLogicalUnit 2 }

  scsiLUIdentifierEntry OBJECT-TYPE
       SYNTAX              ScsiLUIdentifierEntry
       MAX-ACCESS          not-accessible
       STATUS              current
       DESCRIPTION
       "An entry (row) containing parameters applicable to the
       corresponding LUN per logical unit. "
       INDEX { scsiInstIndex, scsiDeviceIndex, scsiLUIndex,
       scsiLUIdIndex }
  ::= { scsiLUIdentifierTable 1 }

  ScsiLUIdentifierEntry ::= SEQUENCE {
       scsiLUIdIndex            ScsiIndexValue,
       scsiLUIdCodeSet          INTEGER,
       scsiLUIdAssociation      INTEGER,
       scsiLUIdType             INTEGER,
       scsiLUIdValue            OCTET STRING
  }

  scsiLUIdIndex  OBJECT-TYPE
       SYNTAX              ScsiIndexValue
       MAX-ACCESS          not-accessible
       STATUS              current
       DESCRIPTION
       "An administrative index for that entry."
  ::= { scsiLUIdentifierEntry 1 }

  scsiLUIdCodeSet OBJECT-TYPE
       SYNTAX              INTEGER {
                                unknown(1),
                                binary(2),
                                ascii(3)
                           }
       MAX-ACCESS          read-only
       STATUS              current
       DESCRIPTION
       "The code set in use for this identifier."
  ::= { scsiLUIdentifierEntry 2 }
  scsiLUIdAssociation OBJECT-TYPE
       SYNTAX             INTEGER {
                               unknown(1),
                               device(2),
                               port(3)
                          }
       MAX-ACCESS         read-only
       STATUS             current
       DESCRIPTION
       "The value device(2) means that the identifier is associated
       with the addressed physical or logical device.
       The value port(3) means that the identifier is associated with
       the port that received the request."
       REFERENCE " ANSI - SCSI Primary Commands - 2 [19]"
       ::= { scsiLUIdentifierEntry 3 }

  scsiLUIdType   OBJECT-TYPE
       SYNTAX             INTEGER {
                               unknown(1),
                               proprietary(2),
                               annexD(3),
                               eui64(4),
                               fcfs(5),
                               relative(6)
                          }
       MAX-ACCESS         read-only
       STATUS             current
       DESCRIPTION
       "This field defines the type of LU Identifier used for this
       identifier and fixes the format of scsiLUIdValue.
       - proprietary(2) corresponds to 0h identifier type value and
       means that the format used for this LU Identifier is vendor-
       specific,
       - annexD(3) corresponds to 1h identifier type value and means
       that this LU Identifier starts with an 8-bytes T10 Vendor ID.
       - eui64(4) corresponds to 2h identifier type value and means
       that the format used for this LU Identifier is IEEE Extended
       Unique Identifier of 64 bits (EUI-64)
       - fcfs(5) corresponds to 3h identifier and means that this LU
       Identifier contains an FC-FS identifier.
       - relative(6) corresponds to 4h identifier type and means that
       this LU Identifier contains the relative position of the port.
       This type is used when the  scsiLUIdAssociation is equal to
       port(3).  "
       REFERENCE " ANSI - SCSI Primary Commands - 2 [19]"
       ::= { scsiLUIdentifierEntry 4 }

  scsiLUIdValue OBJECT-TYPE
       SYNTAX             OCTET STRING (SIZE (0..255))
       MAX-ACCESS         read-only
       STATUS             current
       DESCRIPTION
       "The actual value of this identifier. The format is defined
       by the previous fields."
       ::= { scsiLUIdentifierEntry 5 }

  scsiLUIntrPortTable OBJECT-TYPE
       SYNTAX           SEQUENCE OF ScsiLUIntrPortEntry
       MAX-ACCESS       not-accessible
       STATUS           current
       DESCRIPTION
        "This table includes LUNs additional to the default one. It
       is configured per initiator port"
  ::= { scsiLogicalUnit 3 }

  scsiLUIntrPortEntry OBJECT-TYPE
       SYNTAX              ScsiLUIntrPortEntry
       MAX-ACCESS          not-accessible
       STATUS              current
       DESCRIPTION
       "An entry (row) containing parameters applicable to the
       corresponding LUN per logical unit and per configured
       initiator port. "
       INDEX { scsiInstIndex, scsiDeviceIndex,
       scsiLUIndex,scsiPortIndex, scsiTgtPrtIntrPrtIndex}
  ::= { scsiLUIntrPortTable 1 }

  ScsiLUIntrPortEntry ::= SEQUENCE {
       scsiLUIntrPortLun        ScsiLUNFormat,
       scsiLUIntrPortRowStatus  RowStatus
  }

  scsiLUIntrPortLun   OBJECT-TYPE
       SYNTAX              ScsiLUNFormat
       MAX-ACCESS          read-create
       STATUS              current
       DESCRIPTION
       "It is the LUN that will get the initiator port corresponding
       to the scsiTgtPrtIntrPrtIndex while attached to this logical
       unit."
  ::= { scsiLUIntrPortEntry 1 }

  scsiLUIntrPortRowStatus  OBJECT-TYPE
       SYNTAX              RowStatus
       MAX-ACCESS          read-create
       STATUS              current
       DESCRIPTION
       "This field allows an administrator to create and delete this
       entry."
  ::= { scsiLUIntrPortEntry 2 }

  -- The following groups will be defined in next versions
  -- scsiNotifications OBJECT IDENTIFIER ::= { scsiModule 2 }
  -- scsiConformance OBJECT IDENTIFIER ::= { scsiModule 3 }
```

```
END
```

10. Acknowledgments

This document was produced by the SCSI MIB Working Group.

11. References

[1] Harrington, D., Presuhn, R. and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", RFC 2571, April 1999.

[2] Rose, M. and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", STD 16, RFC 1155, May 1990.

[3] Rose, M. and K. McCloghrie, "Concise MIB Definitions", STD 16, RFC 1212, March 1991.

[4] Rose, M., "A Convention for Defining Traps for use with the SNMP", RFC 1215, March 1991.

[5] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[6] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[7] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose,M. and S. Waldbusser, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

[8] Case, J., Fedor, M., Schoffstall, M. and J. Davin, "Simple Network Management Protocol", STD 15, RFC 1157, May 1990.

[9] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Introduction to Community-based SNMPv2", RFC 1901, January 1996.

[10] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Transport Mappings for Version 2 of the Simple Network Management Protocol(SNMPv2)", RFC 1906, January 1996.

[11] Case, J., Harrington D., Presuhn R. and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", RFC 2572, April 1999.

[12] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM)for version 3 of the Simple Network Management Protocol(SNMPv3)", RFC 2574, April 1999.

[13] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1905, January 1996.

[14] Levi, D., Meyer, P. and B. Stewart, "SNMPv3 Applications", RFC 2573, April 1999.

[15] Wijnen, B., Presuhn, R. and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", RFC 2575, April 1999.

[16] Case, J., Mundy, R., Partain, D. and B. Stewart, "Introduction to Version 3 of the Internet-standard Network Management Framework", RFC 2570, April 1999.

[17] Information Technology, SCSI Architecture Model-2 (SAM-2), Working Draft, T10 Project 1157-D, Revision 20, 19 September 2001

[18] IEEE Tutorial for SCSI use of IEEE company_id - X3T10/97-101, revision 2

[19] Information Technology, SCSI Primary Commands - 2 (SPC-2), T10 Project 1236-D, Revision 20, 18 July 2001

[20] Information Technology, Names, Addresses, Identifiers, Oh my!, T10 Project, Revision 4, 25 July 2001 (T10/01-084 revision 4)

[22] S. Waldbusser and P. Grillo, "Host Resources MIB", RFC 2790, March 2000.

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

12. Security Considerations

There are a number of management objects defined in this MIB 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.

Some managed objects in this MIB may contain sensitive information.

SNMPv1 by itself is not a secure environment. 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.

It is recommended that the implementers consider the security features as provided by the SNMPv3 framework. Specifically, the use of the User-based Security Model RFC 2574 [12] and the View-based Access Control Model RFC 2575 [15] are recommended.

It is then a customer/user responsibility to ensure that the SNMP entity giving access to an instance of this MIB, 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.

13. Authors' Addresses

Michele Hallak-Stamler
Sanrad Intelligent Network
32 Habarzel Street
Tel Aviv, Israel

Phone: +972 3 7674809
Email: michele@sanrad.com

Yaron Lederman
Siliquent Technologies Ltd.
33 Bezalel Street
Ramat Gan, Israel

Phone: +972 3 7552320
Email: yaronl@siliquent.com