Management Information Base for
the Session Initiation Protocol (SIP)

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

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 a set of managed objects that are used to
manage Session Initiation Protocol (SIP) entities, which include User
Agents, and Proxy, Redirect and Registrar servers.

Table of Contents

1. Introduction

   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 that are used to
   manage Session Initiation Protocol (SIP) entities, which include User
   Agents, and Proxy, Redirect and Registrar servers. The managed
   objects defined in this document are intended to provide basic SIP
   protocol management for SIP entities. The management of
   application-specific or service-specific SIP configuration is out of
   scope.

2. Conventions

   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 RFC 2119 [RFC2119].

3. 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 set
of MIB modules that are compliant to the SMIv2, which is described in
STD 58, comprised of RFC 2578 [RFC2578], RFC 2579 [RFC2579], and RFC
2580 [RFC2580].

4. Overview

SIP [RFC3261] is an application-layer control (signaling) protocol
for creating, modifying, and terminating sessions with one or more
participants. These sessions include Internet telephone calls,
multimedia distribution, and multimedia conferences.

This MIB provides some managed objects for SIP entities defined in
RFC 3261 [RFC3261] - User Agents (UA), and Proxy, Redirect, and
Registrar servers. It is intended to provide management of the basic
SIP entities. It provides for monitoring of status and protocol
statistics, as well as for configuration of SIP entities.

5. Structure of the SIP MIB

Four MIB modules are specified: SIP-COMMON-MIB, SIP-SERVER-MIB, SIP-
UA-MIB, and SIP-TC-MIB. SIP-COMMON-MIB contains common MIB objects
used in all the SIP entities. SIP-SERVER-MIB contains objects
specific to Proxy, Redirect, and Registrar servers. SIP-UA-MIB
includes objects specific to User Agents. SIP-TC-MIB defines the
textual conventions used throughout the MIB modules.

The MIB modules contain the following groups of objects:

SIP-COMMON-MIB: Management objects common to all the SIP entities

o sipCommonMIBObjects

   * sipCommonCfgBase: This object group defines configuration
      objects common to all SIP entities, including the SIP protocol
      version, the type of SIP entity (UA, proxy, redirect, registrar
      servers), the operational and administrative status, the SIP
      organization name, the maximum number of SIP transactions an
      entity can manage, etc.

   * sipCommonCfgTimer: This object group defines timer
      configuration objects applicable to SIP user agent and stateful
      SIP proxy entities.

* **sipCommonSummaryStats**: This object group defines a table containing the summary statistics objects applicable to all SIP entities, including the total number of all SIP requests and responses in/out and the total number of transactions.

* **sipCommonMethodStats**: This object group defines a table containing the SIP method statistics objects applicable to all SIP entities, including the number of outbound and inbound requests on a per method basis. Retransmissions, where appropriate, are also included in these statistics.

* **sipCommonStatusCode**: This object group defines a table indicating the number of SIP responses (in and out) that the SIP entity has been requested to monitor on a per-method basis (100, 200, 302, etc.).

* **sipCommonStatsTrans**: This object group defines a table containing a gauge reflecting the number of transactions currently awaiting definitive responses by the managed SIP entity.

* **sipCommonStatsRetry**: This object group defines statistic objects indicating the number of retransmissions sent on a per-method basis.

* **sipCommonOtherStats**: This object group defines additional statistics objects including the number of SIP requests received with unsupported URIs, the number of requests received with unsupported SIP methods, and the number of discarded messages.

* **sipCommonNotifObjects**: This object group defines objects accessible only via a notification (MAX ACCESS clause of accessible-for-notify): they are related to the SNMP notifications defined in this MIB module.

The SIP-COMMON-MIB also contains notifications, including:

o **sipCommonStatusCodeNotif**: indicates that a specific status code has been sent or received by the system.

o **sipCommonStatusCodeThreshExceededNotif**: indicates that a specific status code has been sent or received by the system frequently enough to exceed the configured threshold.

   SIP-SERVER-MIB: Groups of objects for SIP Proxy, Redirect, and
   Registrar servers

   o sipServerMIBObjects

      * sipServerCfg: This object group defines common server
         configuration objects including the SIP server host address.

      * sipServerProxyCfg: This object group defines configuration
         objects for SIP Proxy Servers including the proxy mode of
         operation (stateless, stateful, call stateful), the proxy
         authentication method(s), realm, etc.

      * sipServerProxyStats: This object group defines a table
         containing the statistics objects applicable to SIP Proxy
         Servers. It includes the number of occurrences of unsupported
         options being specified in received Proxy-Require headers.

      * sipServerRegCfg: This object group defines common configuration
         objects for SIP Registrar servers, including the ability to
         accept third-party registrations, such as the maximum
         registration expiry that may be requested by user agents, the
         maximum number of users the registrar can support, the number
         of currently registered users, per contact registration
         information, etc.

      * sipServerRegStats: This object group contains summary
         statistics objects for SIP Registrar servers. Precisely, it
         contains the number of REGISTER requests that have been
         accepted or rejected.

   SIP-UA-MIB: Group of objects for SIP User Agents

   o sipUAMIBObjects

      * sipUACfgServer: This object group specifies SIP server
         configuration objects applicable to SIP user agents including
         the Internet address of the SIP Server the UA uses to register,
         proxy, or redirect calls.

   To conform with this specification, an SNMP agent MUST implement the
   SIP-TC-MIB module, plus the SIP-COMMON-MIB module and one of the SIP
   entity-type-specific MIB modules (SIP-SERVER-MIB or SIP-UA-MIB) as
   applicable for each instance of a SIP entity being managed. If a
   device has more than one SIP entity or multiple instances of the same
   entity type, it MUST implement multiple SIP modules. Section 5.2
   describes handling of multiple instances in detail.

5.1. Textual Conventions

   The data types SipTCTransportProtocol, SipTCEntityRole,
   SipTCOptionTagHeaders, and SipTCMethodName are defined in the SIP-
   TC-MIB module and used as Textual Conventions in this document.

5.2. Relationship to the Network Services MIB

   In the design of the SIP MIB, the authors considered the following
   requirement: the SIP MIB must allow a single system with a single
   SNMP agent to support multiple instances of various SIP MIB modules.
   This requirement is met by using the framework provided by the
   Network Services Monitoring MIB, NETWORK-SERVICES-MIB, RFC 2788
   [RFC2788].

   A device implementing the SIP MIB MUST support the NETWORK-SERVICES-
   MIB and, at a minimum, MUST support the index and name objects
   (applIndex and applName) in the application table (applTable). In
   order to allow each instance of a SIP entity to be managed as a
   separate network service application, a naming convention SHOULD be
   used to make the application name unique. For example, if a system
   is running 2 SIP UAs that need to be managed as 2 separate SIP
   entities, by convention, the application names used in the Network
   Services Monitoring MIB application table should be "sip_ua1" and
   "sip_ua2". This convention allows each instance to have its own row
   in the application table (applTable).

   It is therefore RECOMMENDED that the following application name
   conventions be adopted:

   o for a SIP Proxy entity, the applName value SHOULD be equal to a
      character string starting with "sip_proxy" followed by a unique
      application instance identifier, for example, "sip_proxy1",
      "sip_proxy17".

   o for a SIP Registrar entity, the applName value SHOULD be equal to
      a character string starting with "sip_registrar" followed by a
      unique application instance identifier, for example,
      "sip_registrar1", "sip_registrar2".

   o for a SIP User Agent entity, the applName value SHOULD be equal to
      a character string starting with "sip_ua" followed by a unique
      application instance identifier, for example, "sip_ua1",
      "sip_ua2".

o   for any combination of Proxy, Registrar, or Redirect Server being
    managed as a single aggregate entity, the applName value for the
    combined server entity SHOULD reflect the appropriate combination
    followed by a unique application instance identifier. In order to
    facilitate consistent agent behavior and management application
    expectations, the following order of names is RECOMMENDED:

    *   if Proxy exists, list first.
    *   if Proxy and Redirect exists, list Redirect second.
    *   if Registrar exists, always list last.

    For example "sip_proxy1", "sip_proxy_registrar1",
    "sip_proxy_redirect5", "sip_proxy_redirect_registrar2", or
    "sip_registrar1".

o   Note: the value of the network service application index
    (applIndex) may be different from the instance identifier used in
    the system (the applIndex is dynamically created and is the value
    assigned by the SNMP agent at the creation of the table entry,
    whereas the value of the instance identifier to be used in the
    application name is provided as part of the application name
    applName by the system administrator or configuration files of the
    SIP entity). This note is illustrated in the first example
    provided below.

Finally, the SNMP agent MAY support any combination of the other
attributes in applTable. If supported, the following objects SHOULD
have values populated as follows:

o   applVersion: version of the SIP application.

o   applUptime: the value of applUptime MUST be identical to the value
    of sipCommonCfgServiceStartTime defined in the SIP-COMMON-MIB
    module.

o   applOperStatus: the value of applOperStatus SHOULD reflect the
    operational status of sipCommonCfgServiceOperStatus, at least by
    means of a mapping.

o   applLastChange: the value of applLastChange MUST be identical to
    the value of sipCommonCfgServiceLastChange defined in the SIP-
    COMMON module.

A number of other objects are defined as part of the applTable. They
are not included for the sake of brevity and due to the fact that
they do not enhance the concept being presented.

Example 1:

The tables below illustrate how a system acting as both Proxy and
Registrar server might be configured to maintain separate SIP-
COMMON-MIB instances.

The NETWORK-SERVICES-MIB applTable might be populated as follows:

```
+-----------+-------------------+----------------------+
| applIndex |      applName     |    applDescription   |
+-----------+-------------------+----------------------+
|     1     |    "sip_proxy10"  |    "ACME SIP Proxy"  |
|     2     | "sip_registrar17" | "ACME SIP Registrar" |
+-----------+-------------------+----------------------+
```

The SIP-COMMON-MIB sipCommonCfgTable would have two rows: one for the
proxy (applIndex=1) and one for the registrar (applIndex=2). The
SIP-SERVER-MIB tables would, however, only be populated with one row
indexed by applIndex=1 and applIndex=2, respectively, if the server
provides either proxy or registrar.

The SIP-COMMON-MIB sipCommonCfgTable might be populated as:

```
+---------+------------------------+--------------------------+-----+
|applIndex| sipCommonCfgProtocol   | sipCommonCfgServiceOper   | ... |
|         | Version                | Status                    |     |
+---------+------------------------+--------------------------+-----+
|    1    |        "SIP/2.0"       |          up(1)            |     |
|    2    |        "SIP/2.0"       |       restarting(4)       |     |
+---------+------------------------+--------------------------+-----+
```

while the sipServerProxyCfgTable in SIP-SERVER-MIB might be populated
as:

```
+-----------+-------------------------------+-----+
| applIndex | sipServerCfgProxyStatefulness | ... |
+-----------+-------------------------------+-----+
|     1     |          stateless(1)         |     |
+-----------+-------------------------------+-----+
```

and the sipServerRegUserTable in SIP-SERVER-MIB might be populated
as:

```
+-----------+----------------------+----------------------+-----+
| applIndex | sipServerRegUserIndex | sipServerRegUserUri | ... |
+-----------+----------------------+----------------------+-----+
|     2     |          1           |   bob@example.com    |     |
|     2     |          2           |  alice@example.com   |     |
|     2     |          3           |   jim@example.com    |     |
|     2     |          4           |   john@example.com   |     |
+-----------+----------------------+----------------------+-----+
```

Example 2:

This example illustrates how to represent a system acting as both
Proxy and Registrar server, where the two entities share a single
instance of SIP-COMMON-MIB.

The NETWORK-SERVICES-MIB applTable might be populated as follows:

```
+-----------+----------------------+----------------------------+
| applIndex |       applName       |       applDescription      |
+-----------+----------------------+----------------------------+
|     1     | "sip_proxy_registrar1" |    "ACME SIP Proxy and    |
|           |                      |         Registrar"         |
+-----------+----------------------+----------------------------+
```

The SIP-COMMON-MIB sipCommonCfgTable would have only one row to cover
both the proxy and the registrar.

The SIP-COMMON-MIB sipCommonCfgTable might be populated as:

```
+----------+-------------------------+-----------------------------+
|applIndex |sipCommonCfgProtocolVersion|sipCommonCfgServiceOperStatus|
+----------+-------------------------+-----------------------------+
|    1     |        "SIP/2.0"        |            up(1)            |
+----------+-------------------------+-----------------------------+
```

while the sipServerRegUserTable in SIP-SERVER-MIB might be populated
as:

```
+-----------+---------------------+---------------------+-----+
| applIndex | sipServerRegUserIndex | sipServerRegUserUri | ... |
+-----------+---------------------+---------------------+-----+
|     2     |          1          |   bob@example.com   |     |
|     2     |          2          |  alice@example.com  |     |
|     2     |          3          |  kevin@example.com  |     |
|     2     |          4          |   jf@example.com    |     |
+-----------+---------------------+---------------------+-----+
```

The NETWORK-SERVICES-MIB assocTable is not considered a requirement
for SIP systems. It is not a mandatory group for compliance with the
NETWORK-SERVICES-MIB module.

The relationship between the value of applOperStatus and
sipCommonCfgServiceOperStatus is as follows:

```
+-------------------------------+--------------+-------------------+
|  sipCommonCfgServiceOperStatus |      --      |   applOperStatus  |
|                               |  corresponds |                   |
|                               |    to -->    |                   |
+-------------------------------+--------------+-------------------+
|              up               |     -->      |        up         |
|             down              |     -->      |       down        |
|           congested           |     -->      |     congested     |
|           restarting          |     -->      |     restarting    |
|           quiescing           |     -->      |     quiescing     |
|            testing            |     -->      |        up         |
|            unknown            |     -->      | --indeterminate-- |
+-------------------------------+--------------+-------------------+
```

If the sipOperStatus is 'unknown', there is no corresponding value of
applOperStatus. Therefore, the last known value of applOperStatus
SHOULD be maintained until the sipOperStatus transitions to a value
that can be mapped appropriately.

5.3. IMPORTed MIB Modules and REFERENCE Clauses

The SIP MIB modules defined in this document IMPORT definitions
normatively from the following MIB modules, beyond [RFC2578],
[RFC2579], and [RFC2580]: INET-ADDRESS-MIB [RFC4001], NETWORK-
SERVICES-MIB [RFC2788], SNMP-FRAMEWORK-MIB [RFC3411].

This MIB module also includes REFERENCE clauses that normatively
refer to SIP [RFC3261] and INET-ADDRESS-MIB [RFC4001].

Finally, this MIB module makes informative references to several RFCs
in some of the examples described in the DESCRIPTION clauses,
including Reliability of Provisional Responses in SIP [RFC3262] and
SIP over SCTP [RFC4168].

6. Accommodating SIP Extension Methods

The core set of SIP methods is defined in RFC 3261 [RFC3261]. Other
IETF RFCs define additional methods. In the future, additional
methods may be defined. In order to avoid having to update the SIP-
COMMON-MIB module to accommodate these extension methods, we use a
method identifier name (SipTCMethodName TEXTUAL-CONVENTION) to
represent all SIP methods registered with IANA.

For example, the sipCommonMethodSupportedTable is the main table for
listing all of the SIP methods supported by a system, including the
SIP methods defined in RFC 3261 [RFC3261] and other SIP methods
registered with IANA. The table is informational in nature and
populated by the system. Entries cannot be added or deleted by an
SNMP manager.

The SIP specification (RFC 3261 [RFC3261], Section 27.4) establishes
the sub-registries for SIP Methods and Response Codes under
http://www.iana.org/assignments/sip-parameters. This document uses
the existing sub-registry for the names of registered SIP methods.

For example, in the sipCommonMethodSupportedTable of SIP-COMMON-MIB,
the sipCommonMethodSupportedName values can be represented as
follows:

```
+-------------------------------+
| sipCommonMethodSupportedName  |
+-------------------------------+
              "ACK"
              "BYE"
            "CANCEL"
            "INVITE"
           "OPTIONS"
+-------------------------------+
```

7. Definitions

7.1. SIP Textual Conventions

```
SIP-TC-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY,
    mib-2
          FROM SNMPv2-SMI        -- RFC 2578

    TEXTUAL-CONVENTION
          FROM SNMPv2-TC;        -- RFC 2579

sipTC MODULE-IDENTITY
    LAST-UPDATED "200704200000Z"
    ORGANIZATION "IETF Session Initiation Protocol Working Group"
    CONTACT-INFO
            "SIP WG email: sip@ietf.org

             Co-editor  Kevin Lingle
                        Cisco Systems, Inc.
             postal:    7025 Kit Creek Road
                        P.O. Box 14987
                        Research Triangle Park, NC 27709
                        USA
             email:     klingle@cisco.com
             phone:     +1 919 476 2029

             Co-editor  Joon Maeng
             email:     jmaeng@austin.rr.com

             Co-editor  Jean-Francois Mule
                        CableLabs
             postal:    858 Coal Creek Circle
                        Louisville, CO 80027
                        USA
             email:     jf.mule@cablelabs.com
             phone:     +1 303 661 9100

             Co-editor  Dave Walker
             email:     drwalker@rogers.com"
    DESCRIPTION
        "Session Initiation Protocol (SIP) MIB TEXTUAL-CONVENTION
         module used by other SIP-related MIB Modules.

         Copyright (C) The IETF Trust (2007).  This version of
         this MIB module is part of RFC 4780; see the RFC itself for
```

```
            full legal notices."
      REVISION       "200704200000Z"
      DESCRIPTION
        "Initial version of the IETF SIP-TC-MIB module.  This version
         published as part of RFC 4780."
      ::= { mib-2 148 }
```

--
-- Textual Conventions
--

```
SipTCTransportProtocol ::= TEXTUAL-CONVENTION
      STATUS        current
      DESCRIPTION
        "This convention is a bit map.  Each bit represents a transport
         protocol.  If a bit has value 1, then that selected transport
         protocol is in some way dependent on the context of the object
         using this convention.  If a bit has value 0, then that
         transport protocol is not selected.  Combinations of bits can
         be set when multiple transport protocols are selected.

         bit 0: a protocol other than those defined here
         bit 1: User Datagram Protocol
         bit 2: Transmission Control Protocol
         bit 3: Stream Control Transmission Protocol
         bit 4: Transport Layer Security Protocol over TCP
         bit 5: Transport Layer Security Protocol over SCTP
        "
      REFERENCE "RFC 3261, Section 18 and RFC 4168"
      SYNTAX        BITS {
                      other(0),   -- none of the following
                      udp(1),
                      tcp(2),
                      sctp(3),    -- RFC4168
                      tlsTcp(4),
                      tlsSctp(5) -- RFC 4168
                   }

SipTCEntityRole ::= TEXTUAL-CONVENTION
      STATUS        current
      DESCRIPTION
        "This convention defines the role of a SIP entity.  Examples of
         SIP entities are proxies, user agents, redirect servers,
         registrars, or combinations of the above.

         User Agent (UA): A logical entity that can act as both a user
         agent client and user agent server.
```

User Agent Client (UAC): A logical entity that creates a new
request, and then uses the client transaction state machinery
to send it. The role of UAC lasts only for the duration of
that transaction. In other words, if a piece of software
initiates a request, it acts as a UAC for the duration of that
transaction. If it receives a request later, it assumes the
role of a user agent server for the processing of that
transaction.

User Agent Server (UAS): A logical entity that generates a
response to a SIP request. The response accepts, rejects,
or redirects the request. This role lasts only for the
duration of that transaction. In other words, if a piece of
software responds to a request, it acts as a UAS for the
duration of that transaction. If it generates a request
later, it assumes the role of a user agent client for the
processing of that transaction.

Proxy, Proxy Server: An intermediary entity that acts as both
a server and a client for the purpose of making requests on
behalf of other clients. A proxy server primarily plays the
role of routing, which means its job is to ensure that a
request is sent to another entity 'closer' to the targeted
user. Proxies are also useful for enforcing policy. A proxy
interprets and, if necessary, rewrites specific parts of a
request message before forwarding it.

Redirect Server: A redirect server is a user agent server that
generates 3xx responses to requests it receives, directing the
client to contact an alternate set of URIs.

Registrar: A registrar is a server that accepts REGISTER
requests and places the information it receives in those
requests into the location service for the domain it handles."
    REFERENCE
        "RFC 3261, Section 6"
    SYNTAX   BITS {
                    other(0),
                    userAgent(1),
                    proxyServer(2),
                    redirectServer(3),
                    registrarServer(4)
                }

SipTCOptionTagHeaders ::= TEXTUAL-CONVENTION
    STATUS   current
    DESCRIPTION
        "This convention defines the header fields that use the option

```
            tags per Section 19.2 of RFC 3261.  These tags are used in
            Require (Section 20.32), Proxy-Require (Section 20.29),
            Supported (Section 20.37), and Unsupported (Section 20.40)
            header fields."
    REFERENCE
        "RFC 3261, Sections 19.2, 20.32, 20.29, 20.37, and 20.40"
    SYNTAX        BITS {
                    require(0),       -- Require header
                    proxyRequire(1),  -- Proxy-Require header
                    supported(2),     -- Supported header
                    unsupported(3)    -- Unsupported header
                  }

SipTCMethodName ::= TEXTUAL-CONVENTION
    STATUS        current
    DESCRIPTION
        "This TEXTUAL-CONVENTION is a string that uniquely identifies a
        SIP method.  The scope of uniqueness is the context of all
        defined SIP methods.

        Experimental support of extension methods is acceptable and
        expected.  Extension methods are those defined in
        Internet-Draft documents but not yet allocated and
        officially sanctioned by IANA.

        To support experimental extension methods, any object using
        this TEXTUAL-CONVENTION as syntax MAY return/accept a method
        identifier value other than those sanctioned by IANA.  That
        system MUST ensure no collisions with officially assigned
        method names."
    REFERENCE
        "RFC 3261, Section 27.4"
    SYNTAX        OCTET STRING (SIZE (1..100))

END
```

7.2. SIP Common MIB Module

```
SIP-COMMON-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY,
    OBJECT-TYPE,
    NOTIFICATION-TYPE,
    Counter32,
    Gauge32,
    TimeTicks,
    Unsigned32,
```

```
   mib-2
         FROM SNMPv2-SMI              -- RFC 2578

   RowStatus,
   TimeStamp,
   TruthValue
         FROM SNMPv2-TC              -- RFC 2579

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

   SnmpAdminString
         FROM SNMP-FRAMEWORK-MIB     -- RFC 3411

   SipTCTransportProtocol,
   SipTCMethodName,
   SipTCEntityRole,
   SipTCOptionTagHeaders
         FROM SIP-TC-MIB             -- RFC 4780

   applIndex
         FROM NETWORK-SERVICES-MIB   -- RFC 2788

   InetPortNumber
         FROM INET-ADDRESS-MIB;      -- RFC 4001

sipCommonMIB MODULE-IDENTITY
    LAST-UPDATED "200704200000Z"
    ORGANIZATION "IETF Session Initiation Protocol Working Group"
    CONTACT-INFO
            "SIP WG email: sip@ietf.org

             Co-editor  Kevin Lingle
                        Cisco Systems, Inc.
             postal:    7025 Kit Creek Road
                        P.O. Box 14987
                        Research Triangle Park, NC 27709
                        USA
             email:     klingle@cisco.com
             phone:     +1 919 476 2029

             Co-editor  Joon Maeng
             email:     jmaeng@austin.rr.com

             Co-editor  Jean-Francois Mule
                        CableLabs
```

                    postal: 858 Coal Creek Circle
                                Louisville, CO 80027
                                USA
                    email: jf.mule@cablelabs.com
                    phone: +1 303 661 9100

                    Co-editor   Dave Walker
                    email: drwalker@rogers.com"
        DESCRIPTION
            "Session Initiation Protocol (SIP) Common MIB module. This
             module defines objects that may be common to all SIP entities.

             SIP is an application-layer signaling protocol for creating,
             modifying and terminating multimedia sessions with one or more
             participants. These sessions include Internet multimedia
             conferences and Internet telephone calls. SIP is defined in
             RFC 3261 (June 2002).

             This MIB is defined for managing objects that are common to
             SIP User Agents (UAs), Proxy, Redirect, and Registrar servers.
             Objects specific to each of these entities MAY be managed using
             entity specific MIBs defined in other modules.

             Copyright (C) The IETF Trust (2007). This version of
             this MIB module is part of RFC 4780; see the RFC itself for
             full legal notices."
        REVISION   "200704200000Z"
        DESCRIPTION
            "Initial version of the IETF SIP-COMMON-MIB module. This
             version published as part of RFC 4780."
          ::= { mib-2 149 }

-- Top-Level Components of this MIB.
sipCommonMIBNotifications OBJECT IDENTIFIER ::= { sipCommonMIB 0 }
sipCommonMIBObjects   OBJECT IDENTIFIER ::= { sipCommonMIB 1 }
sipCommonMIBConformance   OBJECT IDENTIFIER ::= { sipCommonMIB 2 }

--
-- This MIB contains objects that are common to all SIP entities.
--

-- Common basic configuration
sipCommonCfgBase   OBJECT IDENTIFIER ::= { sipCommonMIBObjects 1 }

-- Protocol timer configuration
sipCommonCfgTimer   OBJECT IDENTIFIER ::= { sipCommonMIBObjects 2 }

-- SIP message summary statistics

```
sipCommonSummaryStats  OBJECT IDENTIFIER ::= { sipCommonMIBObjects 3 }

-- Per method statistics
sipCommonMethodStats   OBJECT IDENTIFIER ::= { sipCommonMIBObjects 4 }

-- Per Status code or status code class statistics
sipCommonStatusCode    OBJECT IDENTIFIER ::= { sipCommonMIBObjects 5 }

-- Transaction statistics
sipCommonStatsTrans    OBJECT IDENTIFIER ::= { sipCommonMIBObjects 6 }

-- Method retry statistics
sipCommonStatsRetry    OBJECT IDENTIFIER ::= { sipCommonMIBObjects 7 }

-- Other statistics
sipCommonOtherStats    OBJECT IDENTIFIER ::= { sipCommonMIBObjects 8 }

-- Accessible-for-notify objects
sipCommonNotifObjects  OBJECT IDENTIFIER ::= { sipCommonMIBObjects 9 }

--
-- Common Configuration Objects
--
sipCommonCfgTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF SipCommonCfgEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "This table contains the common configuration objects applicable
        to all SIP entities."
    ::= { sipCommonCfgBase 1 }

sipCommonCfgEntry OBJECT-TYPE
    SYNTAX      SipCommonCfgEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "A row of common configuration.

        Each row represents objects for a particular SIP entity
        instance present in this system.  applIndex is used to uniquely
        identify these instances of SIP entities and correlate them
        through the common framework of the NETWORK-SERVICES-MIB (RFC
        2788)."
    INDEX { applIndex }
    ::= { sipCommonCfgTable 1 }

SipCommonCfgEntry ::= SEQUENCE {
```

```
            sipCommonCfgProtocolVersion        SnmpAdminString,
            sipCommonCfgServiceOperStatus      INTEGER,
            sipCommonCfgServiceStartTime       TimeTicks,
            sipCommonCfgServiceLastChange      TimeTicks,
            sipCommonCfgOrganization           SnmpAdminString,
            sipCommonCfgMaxTransactions        Unsigned32,
            sipCommonCfgServiceNotifEnable     BITS,
            sipCommonCfgEntityType             SipTCEntityRole
        }

sipCommonCfgProtocolVersion OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object will reflect the version of SIP supported by this
        SIP entity.  It will follow the same format as SIP version
        information contained in the SIP messages generated by this SIP
        entity.  For example, entities supporting SIP version 2 will
        return 'SIP/2.0' as dictated by the standard."
    REFERENCE
        "RFC 3261, Section 7.1"
    ::= { sipCommonCfgEntry 1 }

sipCommonCfgServiceOperStatus OBJECT-TYPE
    SYNTAX        INTEGER {
                    unknown(1),
                    up(2),
                    down(3),
                    congested(4),
                    restarting(5),
                    quiescing(6),
                    testing(7)
                  }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object contains the current operational state of
        the SIP application.

        unknown     : The operational status cannot be determined
                      for some reason.
        up          : The application is operating normally and is
                      processing (receiving and possibly issuing) SIP
                      requests and responses.
        down        : The application is currently unable to process
                      SIP messages.
        congested   : The application is operational but no additional
```

```
                        inbound transactions can be accommodated at the
                        moment.
            restarting : The application is currently unavailable, but it
                         is in the process of restarting and will
                         presumably, soon be able to process SIP messages.
            quiescing  : The application is currently operational
                         but has been administratively put into
                         quiescence mode.  Additional inbound
                         transactions MAY be rejected.
            testing    : The application is currently in test mode
                         and MAY not be able to process SIP messages.

        The operational status values defined for this object are not
        based on any specific information contained in the SIP
        standard."
    ::= { sipCommonCfgEntry 2 }

sipCommonCfgServiceStartTime OBJECT-TYPE
    SYNTAX      TimeTicks
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The value of sysUpTime at the time the SIP entity was last
        started.  If started prior to the last re-initialization of the
        local network management subsystem, then this object contains a
        zero value."
    ::= { sipCommonCfgEntry 3 }

sipCommonCfgServiceLastChange OBJECT-TYPE
    SYNTAX      TimeTicks
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The value of sysUpTime at the time the SIP entity entered its
         current operational state.  If the current state was entered
         prior to the last re-initialization of the local network
         management subsystem, then this object contains a zero value."
    ::= { sipCommonCfgEntry 4 }

sipCommonCfgOrganization OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object contains the organization name that the SIP entity
         inserts into Organization headers of SIP messages processed by
         this system.  If the string is empty, no Organization header is
         to be generated."
```

```
      REFERENCE
         "RFC 3261, Section 20.25"
      ::= { sipCommonCfgEntry 5 }

  sipCommonCfgMaxTransactions OBJECT-TYPE
      SYNTAX       Unsigned32 (1..4294967295)
      MAX-ACCESS  read-only
      STATUS       current
      DESCRIPTION
         "This object indicates the maximum number of simultaneous
          transactions per second that the SIP entity can manage.  In
          general, the value of this object SHOULD reflect a level of
          transaction processing per second that is considered high
          enough to impact the system's CPU and/or memory resources to
          the point of deteriorating SIP call processing but not high
          enough to cause catastrophic system failure."
      ::= { sipCommonCfgEntry 6 }

  sipCommonCfgServiceNotifEnable OBJECT-TYPE
      SYNTAX       BITS {
                      sipCommonServiceColdStart(0),
                      sipCommonServiceWarmStart(1),
                      sipCommonServiceStatusChanged(2)
                   }
      MAX-ACCESS  read-write
      STATUS       current
      DESCRIPTION
         "This object specifies which SIP service related notifications
          are enabled.  Each bit represents a specific notification.  If
          a bit has a value 1, the associated notification is enabled and
          will be generated by the SIP entity at the appropriate time.

          Support for these notifications is OPTIONAL: either none or all
          notification values are supported.  If an implementation does
          not support this object, it should return a 'noSuchObject'
          exception to an SNMP GET operation.  If notifications are
          supported, this object's default value SHOULD reflect
          sipCommonServiceColdStart and sipCommonServiceWarmStart enabled
          and sipCommonServiceStatusChanged disabled.

          This object value SHOULD persist across reboots."
      DEFVAL { { sipCommonServiceColdStart,
                 sipCommonServiceWarmStart } }
      ::= { sipCommonCfgEntry 7 }

  sipCommonCfgEntityType OBJECT-TYPE
      SYNTAX       SipTCEntityRole
      MAX-ACCESS  read-only
```

```
    STATUS        current
    DESCRIPTION
       "This object identifies the list of SIP entities to which this
        row is related.  It is defined as a bit map.  Each bit
        represents a type of SIP entity.  If a bit has value 1, the
        SIP entity represented by this row plays the role of this
        entity type.  If a bit has value 0, the SIP entity represented
        by this row does not act as this entity type.  Combinations
        of bits can be set when the SIP entity plays multiple SIP
        roles."
    ::= { sipCommonCfgEntry 8 }

--
-- Support for multiple ports
--
sipCommonPortTable OBJECT-TYPE
    SYNTAX        SEQUENCE OF SipCommonPortEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
       "This table contains the list of ports that each SIP entity in
        this system is allowed to use.  These ports can be advertised
        using the Contact header in a REGISTER request or response."
    ::= { sipCommonCfgBase 2 }

sipCommonPortEntry OBJECT-TYPE
    SYNTAX        SipCommonPortEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
       "Specification of a particular port.

        Each row represents those objects for a particular SIP entity
        present in this system.  applIndex is used to uniquely identify
        these instances of SIP entities and correlate them through
        the common framework of the NETWORK-SERVICES-MIB (RFC 2788)."
    INDEX { applIndex, sipCommonPort }
    ::= { sipCommonPortTable 1 }

SipCommonPortEntry ::= SEQUENCE {
        sipCommonPort                 InetPortNumber,
        sipCommonPortTransportRcv     SipTCTransportProtocol
    }

sipCommonPort OBJECT-TYPE
    SYNTAX        InetPortNumber (1..65535)
    MAX-ACCESS    not-accessible
    STATUS        current
```

       DESCRIPTION
          "This object reflects a particular port that can be used by the
          SIP application."
       ::= { sipCommonPortEntry 1 }

sipCommonPortTransportRcv OBJECT-TYPE
    SYNTAX   SipTCTransportProtocol
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
       "This object will specify the transport protocol the SIP entity
        will use to receive SIP messages.

        This object is a bit map. Each bit represents a transport
        protocol. If a bit has value 1, then that transport protocol
        is currently being used. If a bit has value 0, then that
        transport protocol is currently not being used."
    ::= { sipCommonPortEntry 2 }

--
-- Support for SIP option tags (SIP extensions).
-- SIP extensions MAY be supported or required by SIP entities.
--

sipCommonOptionTagTable OBJECT-TYPE
    SYNTAX   SEQUENCE OF SipCommonOptionTagEntry
    MAX-ACCESS   not-accessible
    STATUS   current
    DESCRIPTION
       "This table contains a list of the SIP option tags (SIP
        extensions) that are either required, supported, or
        unsupported by the SIP entity. These option tags are
        used in the Require, Proxy-Require, Supported, and
        Unsupported header fields.

        Example: If a user agent client supports, and requires the
        server to support, reliability of provisional responses
        (RFC 3262), this table contains a row with the option tag string
        '100rel' in sipCommonOptionTag and the OCTET STRING value of
        '1010 0000' or '0xA0' in sipCommonOptionTagHeaderField.

        If a server does not support the required feature (indicated in
        a Require header to a UAS, or in a Proxy-Require to a Proxy
        Server), the server returns a 420 Bad Extension listing the
        feature in an Unsupported header.

        Normally, the list of such features supported by an entity is
        static (i.e., will not change over time)."

```
   REFERENCE
      "RFC 3261, Sections 19.2, 20.32, 20.29, 20.37, and 20.40"
   ::= { sipCommonCfgBase 3 }

sipCommonOptionTagEntry OBJECT-TYPE
    SYNTAX       SipCommonOptionTagEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "A particular SIP option tag (extension) supported or
        unsupported by the SIP entity, and which may be supported or
        required by a peer.

        Each row represents those objects for a particular SIP entity
        present in this system.  applIndex is used to uniquely identify
        these instances of SIP entities and correlate them through the
        common framework of the NETWORK-SERVICES-MIB (RFC 2788)."
    INDEX { applIndex, sipCommonOptionTagIndex }
    ::= { sipCommonOptionTagTable 1 }

SipCommonOptionTagEntry ::= SEQUENCE {
        sipCommonOptionTagIndex         Unsigned32,
        sipCommonOptionTag              SnmpAdminString,
        sipCommonOptionTagHeaderField   SipTCOptionTagHeaders
    }

sipCommonOptionTagIndex OBJECT-TYPE
    SYNTAX       Unsigned32 (1..4294967295)
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
      "This object uniquely identifies a conceptual row in the table."
    ::= { sipCommonOptionTagEntry 1 }

sipCommonOptionTag OBJECT-TYPE
    SYNTAX       SnmpAdminString
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
      "This object indicates the SIP option tag.  The option tag names
       are registered with IANA and available at http://www.iana.org."
    REFERENCE "RFC 3261, Section 27.1"
    ::= { sipCommonOptionTagEntry 2 }

sipCommonOptionTagHeaderField OBJECT-TYPE
    SYNTAX       SipTCOptionTagHeaders
    MAX-ACCESS   read-only
    STATUS       current
```

        DESCRIPTION
           "This object indicates whether the SIP option tag is supported
            (Supported header), unsupported (Unsupported header), or
            required (Require or Proxy-Require header) by the SIP entity.
            A SIP option tag may be both supported and required."
        ::= { sipCommonOptionTagEntry 3 }

--
-- Supported SIP Methods
--
sipCommonMethodSupportedTable OBJECT-TYPE
     SYNTAX   SEQUENCE OF SipCommonMethodSupportedEntry
     MAX-ACCESS   not-accessible
     STATUS   current
     DESCRIPTION
        "This table contains a list of methods supported by each SIP
         entity in this system (see the standard set of SIP methods in
         Section 7.1 of RFC 3261). Any additional methods that may be
         incorporated into the SIP protocol can be represented by this
         table without any requirement to update this MIB module.

         The table is informational in nature and conveys capabilities
         of the managed system to the SNMP Manager.

         From a protocol point of view, the list of methods advertised
         by the SIP entity in the Allow header (Section 20.5 of RFC
         3261) MUST be consistent with the methods reflected in this
         table."
 ::= { sipCommonCfgBase 4 }

sipCommonMethodSupportedEntry OBJECT-TYPE
     SYNTAX   SipCommonMethodSupportedEntry
     MAX-ACCESS   not-accessible
     STATUS   current
     DESCRIPTION
        "A particular method supported by the SIP entity.

         Each row represents those objects for a particular SIP entity
         present in this system. applIndex is used to uniquely identify
         these instances of SIP entities and correlate them through
         the common framework of the NETWORK-SERVICES-MIB (RFC 2788)."
     INDEX { applIndex, sipCommonMethodSupportedIndex }
     ::= { sipCommonMethodSupportedTable 1 }

SipCommonMethodSupportedEntry ::= SEQUENCE {
        sipCommonMethodSupportedIndex   Unsigned32,
        sipCommonMethodSupportedName   SipTCMethodName
     }

```
sipCommonMethodSupportedIndex OBJECT-TYPE
    SYNTAX       Unsigned32 (1..4294967295)
    MAX-ACCESS  not-accessible
    STATUS       current
    DESCRIPTION
       "This object uniquely identifies a conceptual row in the table
        and reflects an assigned number used to identify a specific
        SIP method.

        This identifier is suitable for referencing the associated
        method throughout this and other MIBs supported by this managed
        system."
    ::= { sipCommonMethodSupportedEntry 1 }

sipCommonMethodSupportedName OBJECT-TYPE
    SYNTAX       SipTCMethodName
    MAX-ACCESS  read-only
    STATUS       current
    DESCRIPTION
       "This object reflects the supported method's name.  The method
        name MUST be all upper case (e.g., 'INVITE')."
 ::= { sipCommonMethodSupportedEntry 2 }

--
-- SIP Timer Configuration
--
sipCommonCfgTimerTable OBJECT-TYPE
    SYNTAX       SEQUENCE OF SipCommonCfgTimerEntry
    MAX-ACCESS  not-accessible
    STATUS       current
    DESCRIPTION
       "This table contains timer configuration objects applicable to
        SIP user agent and SIP stateful Proxy Server entities."
    ::= { sipCommonCfgTimer 1 }

sipCommonCfgTimerEntry OBJECT-TYPE
    SYNTAX       SipCommonCfgTimerEntry
    MAX-ACCESS  not-accessible
    STATUS       current
    DESCRIPTION
       "A row of timer configuration.

        Each row represents those objects for a particular SIP entity
        present in this system.  applIndex is used to uniquely identify
        these instances of SIP entities and correlate them through
        the common framework of the NETWORK-SERVICES-MIB (RFC 2788).
        The objects in this table entry SHOULD be non-volatile and
        their value SHOULD be kept at reboot."
```

```
    INDEX { applIndex }
    ::= { sipCommonCfgTimerTable 1 }

SipCommonCfgTimerEntry ::= SEQUENCE {
        sipCommonCfgTimerA              Unsigned32,
        sipCommonCfgTimerB              Unsigned32,
        sipCommonCfgTimerC              Unsigned32,
        sipCommonCfgTimerD              Unsigned32,
        sipCommonCfgTimerE              Unsigned32,
        sipCommonCfgTimerF              Unsigned32,
        sipCommonCfgTimerG              Unsigned32,
        sipCommonCfgTimerH              Unsigned32,
        sipCommonCfgTimerI              Unsigned32,
        sipCommonCfgTimerJ              Unsigned32,
        sipCommonCfgTimerK              Unsigned32,
        sipCommonCfgTimerT1             Unsigned32,
        sipCommonCfgTimerT2             Unsigned32,
        sipCommonCfgTimerT4             Unsigned32
    }

sipCommonCfgTimerA OBJECT-TYPE
    SYNTAX      Unsigned32 (100..1000)
    UNITS       "milliseconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object reflects the initial value for the retransmit timer
        for the INVITE method.  The retransmit timer doubles after each
        retransmission, ensuring an exponential backoff in network
        traffic.  This object represents the initial time a SIP entity
        will wait to receive a provisional response to an INVITE before
        resending the INVITE request."
    REFERENCE
        "RFC 3261, Section 17.1.1.2"
    DEFVAL { 500 }
    ::= { sipCommonCfgTimerEntry 1 }

sipCommonCfgTimerB OBJECT-TYPE
    SYNTAX      Unsigned32 (32000..300000)
    UNITS       "milliseconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object reflects the maximum time a SIP entity will wait to
        receive a final response to an INVITE.  The timer is started
        upon transmission of the initial INVITE request."
    REFERENCE
        "RFC 3261, Section 17.1.1.2"
```

```
    DEFVAL { 32000 }
::= { sipCommonCfgTimerEntry 2 }

sipCommonCfgTimerC OBJECT-TYPE
    SYNTAX       Unsigned32 (180000..300000)
    UNITS        "milliseconds"
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "This object reflects the maximum time a SIP Proxy Server will
        wait to receive a provisional response to an INVITE.  The Timer
        C MUST be set for each client transaction when an INVITE
        request is proxied."
    REFERENCE
       "RFC 3261, Section 16.6"
    DEFVAL { 180000 }
    ::= { sipCommonCfgTimerEntry 3 }

sipCommonCfgTimerD OBJECT-TYPE
    SYNTAX       Unsigned32 (0..300000)
    UNITS        "milliseconds"
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "This object reflects the amount of time that the server
        transaction can remain in the 'Completed' state when unreliable
        transports are used.  The default value MUST be equal to or
        greater than 32000 for UDP transport, and its value MUST be 0
        for TCP/SCTP transport."
    REFERENCE
       "RFC 3261, Section 17.1.1.2"
    DEFVAL { 32000 }
    ::= { sipCommonCfgTimerEntry 4 }

sipCommonCfgTimerE OBJECT-TYPE
    SYNTAX       Unsigned32 (100..1000)
    UNITS        "milliseconds"
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "This object reflects the initial value for the retransmit timer
        for a non-INVITE method while in 'Trying' state.  The
        retransmit timer doubles after each retransmission until it
        reaches T2 to ensure an exponential backoff in network traffic.
        This object represents the initial time a SIP entity will wait
        to receive a provisional response to the request before
        resending the non-INVITE request."
    REFERENCE
```

```
        "RFC 3261, Section 17.1.2.2"
    DEFVAL { 500 }
    ::= { sipCommonCfgTimerEntry 5 }

sipCommonCfgTimerF  OBJECT-TYPE
    SYNTAX      Unsigned32 (32000..300000)
    UNITS       "milliseconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object reflects the maximum time a SIP entity will wait to
         receive a final response to a non-INVITE request.  The timer is
         started upon transmission of the initial request."
    REFERENCE
        "RFC 3261, Section 17.1.2.2"
    DEFVAL { 32000 }

    ::= { sipCommonCfgTimerEntry 6 }

sipCommonCfgTimerG  OBJECT-TYPE
    SYNTAX      Unsigned32 (0..1000)
    UNITS       "milliseconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object reflects the initial value for the retransmit timer
         for final responses to INVITE requests.  If timer G fires, the
         response is passed to the transport layer again for
         retransmission, and timer G is set to fire in MIN(2*T1, T2)
         seconds.  From then on, when timer G fires, the response is
         passed to the transport again for transmission, and timer G is
         reset with a value that doubles, unless that value exceeds T2,
         in which case, it is reset with the value of T2.  The default
         value MUST be T1 for UDP transport, and its value MUST be 0 for
         reliable transport like TCP/SCTP."
    REFERENCE
        "RFC 3261, Section 17.2.1"
    DEFVAL { 500 }
    ::= { sipCommonCfgTimerEntry 7 }

sipCommonCfgTimerH  OBJECT-TYPE
    SYNTAX      Unsigned32 (32000..300000)
    UNITS       "milliseconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object reflects the maximum time a server will wait to
         receive an ACK before it abandons retransmitting the response.
```

            The timer is started upon entering the 'Completed' state."
        REFERENCE
           "RFC 3261, Section 17.2.1"
        DEFVAL { 32000 }
        ::= { sipCommonCfgTimerEntry 8 }

    sipCommonCfgTimerI OBJECT-TYPE
        SYNTAX   Unsigned32 (0..10000)
        UNITS   "milliseconds"
        MAX-ACCESS   read-only
        STATUS   current
        DESCRIPTION
           "This object reflects the maximum time a SIP entity will wait to
            receive additional ACK message retransmissions.

            The timer is started upon entering the 'Confirmed' state. The
            default value MUST be T4 for UDP transport and its value MUST
            be 0 for reliable transport like TCP/SCTP."
        REFERENCE
           "RFC 3261, Section 17.2.1"
        DEFVAL { 5000 }
        ::= { sipCommonCfgTimerEntry 9 }

    sipCommonCfgTimerJ OBJECT-TYPE
        SYNTAX   Unsigned32 (32000..300000)
        UNITS   "milliseconds"
        MAX-ACCESS   read-only
        STATUS   current
        DESCRIPTION
           "This object reflects the maximum time a SIP server will wait to
            receive retransmissions of non-INVITE requests. The timer is
            started upon entering the 'Completed' state for non-INVITE
            transactions. When timer J fires, the server MUST transition to
            the 'Terminated' state."
        REFERENCE
           "RFC 3261, Section 17.2.2"
        DEFVAL { 32000 }
        ::= { sipCommonCfgTimerEntry 10 }

    sipCommonCfgTimerK OBJECT-TYPE
        SYNTAX   Unsigned32 (0..10000)
        UNITS   "milliseconds"
        MAX-ACCESS   read-only
        STATUS   current
        DESCRIPTION
           "This object reflects the maximum time a SIP client will wait to
            receive retransmissions of responses to non-INVITE requests.
            The timer is started upon entering the 'Completed' state for

```
              non-INVITE transactions.  When timer K fires, the server MUST
              transition to the 'Terminated' state.  The default value MUST
              be T4 for UDP transport, and its value MUST be 0 for reliable
              transport like TCP/SCTP."
       REFERENCE
          "RFC 3261, Section 17.1.2.2"
       DEFVAL { 5000 }
       ::= { sipCommonCfgTimerEntry 11 }

   sipCommonCfgTimerT1  OBJECT-TYPE
       SYNTAX       Unsigned32 (200..10000)
       UNITS        "milliseconds"
       MAX-ACCESS   read-only
       STATUS       current
       DESCRIPTION
          "This object reflects the T1 timer for a SIP entity.  T1 is an
           estimate of the round-trip time (RTT) between the client and
           server transactions."
       REFERENCE
          "RFC 3261, Section 17"
       DEFVAL { 500 }
       ::= { sipCommonCfgTimerEntry 12 }

   sipCommonCfgTimerT2  OBJECT-TYPE
       SYNTAX       Unsigned32 (200..10000)
       UNITS        "milliseconds"
       MAX-ACCESS   read-only
       STATUS       current
       DESCRIPTION
          "This object reflects the T2 timer for a SIP entity.  T2 is the
           maximum retransmit interval for non-INVITE requests and INVITE
           responses.  It's used in various parts of the protocol to reset
           other Timer* objects to this value."
       REFERENCE
          "RFC 3261, Section 17"
       DEFVAL { 4000 }
       ::= { sipCommonCfgTimerEntry 13 }

   sipCommonCfgTimerT4  OBJECT-TYPE
       SYNTAX       Unsigned32 (200..10000)
       UNITS        "milliseconds"
       MAX-ACCESS   read-only
       STATUS       current
       DESCRIPTION
          "This object reflects the T4 timer for a SIP entity.  T4 is the
           maximum duration a message will remain in the network.  It
           represents the amount of time the network will take to clear
           messages between client and server transactions.  It's used in
```

          various parts of the protocol to reset other Timer* objects to
          this value."
       REFERENCE
          "RFC 3261, Section 17"
       DEFVAL { 5000 }
       ::= { sipCommonCfgTimerEntry 14 }

--
-- Common Statistics Objects
--

--
-- Summary Statistics
--
sipCommonSummaryStatsTable OBJECT-TYPE
     SYNTAX   SEQUENCE OF SipCommonSummaryStatsEntry
     MAX-ACCESS   not-accessible
     STATUS   current
     DESCRIPTION
        "This table contains the summary statistics objects applicable
         to all SIP entities. Each row represents those objects for a
         particular SIP entity present in this system."
     ::= { sipCommonSummaryStats 1 }

sipCommonSummaryStatsEntry OBJECT-TYPE
     SYNTAX   SipCommonSummaryStatsEntry
     MAX-ACCESS   not-accessible
     STATUS   current
     DESCRIPTION
        "A row of summary statistics.

         Each row represents those objects for a particular SIP entity
         present in this system. applIndex is used to uniquely identify
         these instances of SIP entities and correlate them through
         the common framework of the NETWORK-SERVICES-MIB (RFC 2788)."
     INDEX { applIndex }
     ::= { sipCommonSummaryStatsTable 1 }

SipCommonSummaryStatsEntry ::= SEQUENCE {
        sipCommonSummaryInRequests   Counter32,
        sipCommonSummaryOutRequests   Counter32,
        sipCommonSummaryInResponses   Counter32,
        sipCommonSummaryOutResponses   Counter32,
        sipCommonSummaryTotalTransactions   Counter32,
        sipCommonSummaryDisconTime   TimeStamp
     }

sipCommonSummaryInRequests OBJECT-TYPE

```
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "This object indicates the total number of SIP request messages
        received by the SIP entity, including retransmissions.

        Discontinuities in the value of this counter can occur at
        re-initialization of the SIP entity or service.  A Management
        Station can detect discontinuities in this counter by
        monitoring the sipCommonSummaryDisconTime object in the same
        row."
    ::= { sipCommonSummaryStatsEntry 1 }

sipCommonSummaryOutRequests OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "This object contains the total number of SIP request messages
        sent out (originated and relayed) by the SIP entity.  Where a
        particular message is sent more than once, for example as a
        retransmission or as a result of forking, each transmission is
        counted separately.

        Discontinuities in the value of this counter can occur at
        re-initialization of the SIP entity or service.  A Management
        Station can detect discontinuities in this counter by
        monitoring the sipCommonSummaryDisconTime object in the same
        row."
    ::= { sipCommonSummaryStatsEntry 2 }

sipCommonSummaryInResponses OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "This object contains the total number of SIP response messages
        received by the SIP entity, including retransmissions.

        Discontinuities in the value of this counter can occur at
        re-initialization of the SIP entity or service.  A Management
        Station can detect discontinuities in this counter by
        monitoring the sipCommonSummaryDisconTime object in the same
        row."
    ::= { sipCommonSummaryStatsEntry 3 }

sipCommonSummaryOutResponses OBJECT-TYPE
```

```
    SYNTAX       Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "This object contains the total number of SIP response messages
         sent (originated and relayed) by the SIP entity including
         retransmissions.

         Discontinuities in the value of this counter can occur at
         re-initialization of the SIP entity or service.  A Management
         Station can detect discontinuities in this counter by
         monitoring the sipCommonSummaryDisconTime object in the same
         row."
    ::= { sipCommonSummaryStatsEntry 4 }

sipCommonSummaryTotalTransactions OBJECT-TYPE
    SYNTAX       Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "This object contains a count of the number of transactions that
         are in progress and transactions that have reached the
         'Terminated' state.  It is not applicable to stateless SIP Proxy
         Servers.

         A SIP transaction occurs between a client and a server, and
         comprises all messages from the first request sent from the
         client to the server, up to a final (non-1xx) response sent
         from the server to the client.

         If the request is INVITE and the final response is a non-2xx,
         the transaction also include an ACK to the response.  The ACK
         for a 2xx response to an INVITE request is a separate
         transaction.

         The branch ID parameter in the Via header field values serves
         as a transaction identifier.

         A transaction is identified by the CSeq sequence number within
         a single call leg.  The ACK request has the same CSeq number as
         the corresponding INVITE request, but comprises a transaction
         of its own.

         In the case of a forked request, each branch counts as a single
         transaction.

         For a transaction stateless Proxy Server, this counter is
         always 0.
```

          Discontinuities in the value of this counter can occur at
          re-initialization of the SIP entity or service. A Management
          Station can detect discontinuities in this counter by
          monitoring the sipCommonSummaryDisconTime object in the same
          row."
     ::= { sipCommonSummaryStatsEntry 5 }

sipCommonSummaryDisconTime OBJECT-TYPE
     SYNTAX   TimeStamp
     MAX-ACCESS read-only
     STATUS   current
     DESCRIPTION
        "The value of the sysUpTime object when the counters for the
         summary statistics objects in this row last experienced a
         discontinuity."
     ::= { sipCommonSummaryStatsEntry 6 }

--
-- SIP Method Statistics
-- Total counts for each SIP method.
--
sipCommonMethodStatsTable OBJECT-TYPE
     SYNTAX   SEQUENCE OF SipCommonMethodStatsEntry
     MAX-ACCESS not-accessible
     STATUS   current
     DESCRIPTION
        "This table contains the method statistics objects for SIP
         entities. Each row represents those objects for a particular
         SIP entity present in this system."
     ::= { sipCommonMethodStats 1 }

sipCommonMethodStatsEntry OBJECT-TYPE
     SYNTAX   SipCommonMethodStatsEntry
     MAX-ACCESS not-accessible
     STATUS   current
     DESCRIPTION
        "A row of per entity method statistics.

         Each row represents those objects for a particular SIP entity
         present in this system. applIndex is used to uniquely identify
         these instances of SIP entities and correlate them through
         the common framework of the NETWORK-SERVICES-MIB (RFC 2788)."
     INDEX { applIndex, sipCommonMethodStatsName }
     ::= { sipCommonMethodStatsTable 1 }

SipCommonMethodStatsEntry ::= SEQUENCE {
        sipCommonMethodStatsName   SipTCMethodName,
        sipCommonMethodStatsOutbounds   Counter32,

```
        sipCommonMethodStatsInbounds       Counter32,
        sipCommonMethodStatsDisconTime     TimeStamp

    }
```

sipCommonMethodStatsName OBJECT-TYPE
    SYNTAX   SipTCMethodName
    MAX-ACCESS   not-accessible
    STATUS   current
    DESCRIPTION
        "This object uniquely identifies the SIP method related to the
        objects in a particular row."
    ::= { sipCommonMethodStatsEntry 1 }

sipCommonMethodStatsOutbounds OBJECT-TYPE
    SYNTAX   Counter32
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
        "This object reflects the total number of requests sent by the
        SIP entity, excluding retransmissions. Retransmissions are
        counted separately and are not reflected in this counter. A
        Management Station can detect discontinuities in this counter
        by monitoring the sipCommonMethodStatsDisconTime object in the
        same row."
    REFERENCE
        "RFC 3261, Section 7.1"
    ::= { sipCommonMethodStatsEntry 2 }

sipCommonMethodStatsInbounds OBJECT-TYPE
    SYNTAX   Counter32
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
        "This object reflects the total number of requests received by
        the SIP entity. Retransmissions are counted separately and are
        not reflected in this counter. A Management Station can detect
        discontinuities in this counter by monitoring the
        sipCommonMethodStatsDisconTime object in the same row."
    REFERENCE
        "RFC 3261, Section 7.1"
    ::= { sipCommonMethodStatsEntry 3 }

sipCommonMethodStatsDisconTime OBJECT-TYPE
    SYNTAX   TimeStamp
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION

```

         "The value of the sysUpTime object when the counters for the
          method statistics objects in this row last experienced a
          discontinuity."
      ::= { sipCommonMethodStatsEntry 4 }


--
-- Support for specific status codes
--
sipCommonStatusCodeTable OBJECT-TYPE
    SYNTAX        SEQUENCE OF SipCommonStatusCodeEntry
    MAX-ACCESS  not-accessible
    STATUS        current
    DESCRIPTION
       "This table contains the list of SIP status codes that each SIP
        entity in this system has been requested to monitor.  It is the
        mechanism by which specific status codes are monitored.
        Entries created in this table must not persist across reboots."
      ::= { sipCommonStatusCode 1 }

sipCommonStatusCodeEntry OBJECT-TYPE
    SYNTAX        SipCommonStatusCodeEntry
    MAX-ACCESS  not-accessible
    STATUS        current
    DESCRIPTION
       "This row contains information on a particular SIP status code
        that the SIP entity has been requested to monitor.  Entries
        created in this table must not persist across reboots.

        Each row represents those objects for a particular SIP entity
        present in this system.  applIndex is used to uniquely identify
        these instances of SIP entities and correlate them through
        the common framework of the NETWORK-SERVICES-MIB (RFC 2788)."
    INDEX { applIndex, sipCommonStatusCodeMethod,
          sipCommonStatusCodeValue }
      ::= { sipCommonStatusCodeTable 1 }

SipCommonStatusCodeEntry ::= SEQUENCE {
        sipCommonStatusCodeMethod      SipTCMethodName,
        sipCommonStatusCodeValue       Unsigned32,
        sipCommonStatusCodeIns         Counter32,
        sipCommonStatusCodeOuts        Counter32,
        sipCommonStatusCodeRowStatus   RowStatus,
        sipCommonStatusCodeDisconTime  TimeStamp
    }

sipCommonStatusCodeMethod OBJECT-TYPE
    SYNTAX        SipTCMethodName
    MAX-ACCESS  not-accessible

```
    STATUS   current
    DESCRIPTION
      "This object uniquely identifies a conceptual row in the
      table."
    ::= { sipCommonStatusCodeEntry 1 }

sipCommonStatusCodeValue OBJECT-TYPE
    SYNTAX   Unsigned32 (100..999)
    MAX-ACCESS   not-accessible
    STATUS   current
    DESCRIPTION
      "This object contains a SIP status code value that the SIP
       entity has been requested to monitor. All of the other
       information in the row is related to this value."
    ::= { sipCommonStatusCodeEntry 2 }

sipCommonStatusCodeIns OBJECT-TYPE
    SYNTAX   Counter32
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
      "This object reflects the total number of response messages
       received by the SIP entity with the status code value contained
       in the sipCommonStatusCodeValue column.

       Discontinuities in the value of this counter can occur at
       re-initialization of the SIP entity or service, or when the
       monitoring of the status code is temporarily disabled. A
       Management Station can detect discontinuities in this counter
       by monitoring the sipCommonStatusCodeDisconTime object in the
       same row."
    ::= { sipCommonStatusCodeEntry 3 }

sipCommonStatusCodeOuts OBJECT-TYPE
    SYNTAX   Counter32
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
      "This object reflects the total number of response messages sent
       by the SIP entity with the status code value contained in the
       sipCommonStatusCodeValue column.

       Discontinuities in the value of this counter can occur at
       re-initialization of the SIP entity or service, or when the
       monitoring of the Status code is temporarily disabled. A
       Management Station can detect discontinuities in this counter
       by monitoring the sipCommonStatusCodeDisconTime object in the
       same row."
```

```
    ::= { sipCommonStatusCodeEntry 4 }

sipCommonStatusCodeRowStatus OBJECT-TYPE
    SYNTAX       RowStatus
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
       "The row augmentation in sipCommonStatusCodeNotifTable will be
        governed by the value of this RowStatus.

        The values 'createAndGo' and 'destroy' are the only valid
        values allowed for this object.  If a row exists, it will
        reflect a status of 'active' when queried."
    ::= { sipCommonStatusCodeEntry 5 }

sipCommonStatusCodeDisconTime  OBJECT-TYPE
    SYNTAX TimeStamp
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
       "The value of the sysUpTime object when the counters for the
        status code statistics objects in this row last experienced
        a discontinuity."
    ::= { sipCommonStatusCodeEntry 6 }

--
-- Support for specific status code notifications
--

sipCommonStatusCodeNotifTable OBJECT-TYPE
    SYNTAX       SEQUENCE OF SipCommonStatusCodeNotifEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "This table contains objects to control notifications related to
        particular status codes that each SIP entity in this system has
        been requested to monitor.

        There is an entry in this table corresponding to each entry in
        sipCommonStatusCodeTable.  Therefore, this table augments
        sipCommonStatusCodeTable and utilizes the same index
        methodology.

        The objects in this table are not included directly in the
        sipCommonStatusCodeTable simply to keep the status code
        notification control objects separate from the actual status
        code statistics."
    ::= { sipCommonStatusCode 2 }
```

```
sipCommonStatusCodeNotifEntry OBJECT-TYPE
    SYNTAX       SipCommonStatusCodeNotifEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "This row contains information controlling notifications for a
        particular SIP status code that the SIP entity has been
        requested to monitor."
    AUGMENTS { sipCommonStatusCodeEntry }
    ::= { sipCommonStatusCodeNotifTable 1 }

SipCommonStatusCodeNotifEntry ::= SEQUENCE {
        sipCommonStatusCodeNotifSend        TruthValue,
        sipCommonStatusCodeNotifEmitMode    INTEGER,
        sipCommonStatusCodeNotifThresh      Unsigned32,
        sipCommonStatusCodeNotifInterval    Unsigned32
    }

sipCommonStatusCodeNotifSend OBJECT-TYPE
    SYNTAX       TruthValue
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
       "This object controls whether a sipCommonStatusCodeNotif is
        emitted when the status code value specified by
        sipCommonStatusCodeValue is sent or received.  If the value of
        this object is 'true', then a notification is sent.  If it is
        'false', no notification is sent.
        Note well that a notification MAY be emitted for every message
        sent or received that contains the particular status code.
        Depending on the status code involved, this can cause a
        significant number of notification emissions that could be
        detrimental to network performance.  Managers are forewarned to
        be prudent in the use of this object to enable notifications.
        Look to sipCommonStatusCodeNotifEmitMode for alternative
        controls for sipCommonStatusCodeNotif emissions."
    DEFVAL { false }
    ::= { sipCommonStatusCodeNotifEntry 1 }

sipCommonStatusCodeNotifEmitMode OBJECT-TYPE
    SYNTAX         INTEGER {
                     normal(1),
                     oneShot(2),
                     triggered(3)  -- read-only
                   }
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
```

        "The object sipCommonStatusCodeNotifSend MUST be set to 'true'
         for the values of this object to have any effect. It is
         RECOMMENDED that the desired emit mode be established by this
         object prior to setting sipCommonStatusCodeNotifSend to 'true'.
         This object and the sipCommonStatusCodeNotifSend object can
         obviously be set independently, but their respective values
         will have a dependency on each other and the resulting
         notifications.

         This object specifies the mode for emissions of
         sipCommonStatusCodeNotif notifications.

         normal   : sipCommonStatusCodeNotif notifications will be
                      emitted by the system for each SIP response
                      message sent or received that contains the
                      desired status code.

         oneShot   : Only one sipCommonStatusCodeNotif notification
                      will be emitted. It will be the next SIP response
                      message sent or received that contains the
                      desired status code.

                      No more notifications are emitted until this
                      object is set to 'oneShot' again or set to
                      'normal'. This option is provided as a means of
                      quelling the potential promiscuous behavior that
                      can be associated with the
                      sipCommonStatusCodeNotif.

         triggered : This value is only readable and cannot be set. It
                      reflects that the 'oneShot' case has occurred,
                      and indicates that the mode needs to be reset to
                      get further notifications. The mode is reset by
                      setting this object to 'oneShot' or 'normal'."
    DEFVAL { oneShot }
    ::= { sipCommonStatusCodeNotifEntry 2 }

sipCommonStatusCodeNotifThresh OBJECT-TYPE
    SYNTAX   Unsigned32
    MAX-ACCESS   read-write
    STATUS   current
    DESCRIPTION
       "This object specifies the number of response messages sent or
        received by this system that are considered excessive. Based
        on crossing that threshold, a
        sipCommonStatusCodeThreshExceededInNotif notification or a
        sipCommonStatusCodeThreshExceededOutNotif will be sent. The
        sipCommonStatusCodeThreshExceededInNotif and

        sipCommonStatusCodeThreshExceededOutNotif notifications can be
        used as an early warning mechanism in lieu of using
        sipCommonStatusCodeNotif.

        Note that the configuration applied by this object will be
        applied equally to inbound and outbound response messages."
    DEFVAL { 500 }
    ::= { sipCommonStatusCodeNotifEntry 3 }

sipCommonStatusCodeNotifInterval OBJECT-TYPE
    SYNTAX   Unsigned32
    UNITS   "seconds"
    MAX-ACCESS read-write
    STATUS   current
    DESCRIPTION
        "This object specifies the time interval over which, if
        sipCommonStatusCodeThresh is exceeded with respect to sent or
        received messages, a sipCommonStatusCodeThreshExceededInNotif
        or sipCommonStatusCodeThreshExceededOutNotif notification will
        be sent.

        Note that the configuration applied by this object will be
        applied equally to inbound and outbound response messages."
    DEFVAL { 60 }
    ::= { sipCommonStatusCodeNotifEntry 4 }

--
-- Transaction Statistics
--
sipCommonTransCurrentTable OBJECT-TYPE
    SYNTAX   SEQUENCE OF SipCommonTransCurrentEntry
    MAX-ACCESS not-accessible
    STATUS   current
    DESCRIPTION
        "This table contains information on the transactions currently
        awaiting definitive responses by each SIP entity in this
        system.

        This table does not apply to transaction stateless Proxy
        Servers."
    ::= { sipCommonStatsTrans 1 }

sipCommonTransCurrentEntry OBJECT-TYPE
    SYNTAX   SipCommonTransCurrentEntry
    MAX-ACCESS not-accessible
    STATUS   current
    DESCRIPTION
        "Information on a particular SIP entity's current transactions.

          Each row represents those objects for a particular SIP entity
          present in this system. applIndex is used to uniquely identify
          these instances of SIP entities and correlate them through
          the common framework of the NETWORK-SERVICES-MIB (RFC 2788)."
     INDEX { applIndex }
     ::= { sipCommonTransCurrentTable 1 }

SipCommonTransCurrentEntry ::= SEQUENCE {
          sipCommonTransCurrentactions Gauge32
     }

sipCommonTransCurrentactions OBJECT-TYPE
     SYNTAX   Gauge32 (0..4294967295)
     MAX-ACCESS   read-only
     STATUS   current
     DESCRIPTION
        "This object contains the number of transactions awaiting
         definitive (non-1xx) response. In the case of a forked
         request, each branch counts as a single transaction
         corresponding to the entity identified by applIndex."
::= { sipCommonTransCurrentEntry 1 }

--
-- SIP Retry Statistics
--
-- This group contains various statistics objects about
-- retransmission counts.
--
sipCommonStatsRetryTable OBJECT-TYPE
     SYNTAX   SEQUENCE OF SipCommonStatsRetryEntry
     MAX-ACCESS   not-accessible
     STATUS   current
     DESCRIPTION
        "This table contains retry statistics objects applicable to each
         SIP entity in this system."
     ::= { sipCommonStatsRetry 1 }

sipCommonStatsRetryEntry OBJECT-TYPE
     SYNTAX   SipCommonStatsRetryEntry
     MAX-ACCESS   not-accessible
     STATUS   current
     DESCRIPTION
        "A row of retry statistics.

         Each row represents those objects for a particular SIP entity
         present in this system. applIndex is used to uniquely identify
         these instances of SIP entities and correlate them through the
         common framework of the NETWORK-SERVICES-MIB (RFC 2788)."

```
     INDEX { applIndex, sipCommonStatsRetryMethod }
     ::= { sipCommonStatsRetryTable 1 }

SipCommonStatsRetryEntry ::= SEQUENCE {
        sipCommonStatsRetryMethod              SipTCMethodName,
        sipCommonStatsRetries                  Counter32,
        sipCommonStatsRetryFinalResponses      Counter32,
        sipCommonStatsRetryNonFinalResponses   Counter32,
        sipCommonStatsRetryDisconTime          TimeStamp
     }

sipCommonStatsRetryMethod OBJECT-TYPE
    SYNTAX      SipTCMethodName
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
      "This object uniquely identifies the SIP method related to the
       objects in a row."
    ::= { sipCommonStatsRetryEntry 1 }

sipCommonStatsRetries OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
      "This object reflects the total number of request
       retransmissions that have been sent by the SIP entity.  Note
       that there could be multiple retransmissions per request.

       Discontinuities in the value of this counter can occur at
       re-initialization of the SIP entity or service.  A Management
       Station can detect discontinuities in this counter by
       monitoring the sipCommonStatsRetryDisconTime object in the same
       row."
    ::= { sipCommonStatsRetryEntry 2 }

sipCommonStatsRetryFinalResponses OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
      "This object reflects the total number of Final Response retries
       that have been sent by the SIP entity.  Note that there could
       be multiple retransmissions per request.

       Discontinuities in the value of this counter can occur at
       re-initialization of the SIP entity or service.  A Management
       Station can detect discontinuities in this counter by
```

```
         monitoring the sipCommonStatsRetryDisconTime object in the same
         row."
    ::= { sipCommonStatsRetryEntry 3 }

sipCommonStatsRetryNonFinalResponses OBJECT-TYPE
    SYNTAX        Counter32
    MAX-ACCESS  read-only
    STATUS        current
    DESCRIPTION
       "This object reflects the total number of non-Final Response
        retries that have been sent by the SIP entity.

        Discontinuities in the value of this counter can occur at
        re-initialization of the SIP entity or service.  A Management
        Station can detect discontinuities in this counter by
        monitoring the sipCommonStatsRetryDisconTime object in the same
        row."
    ::= { sipCommonStatsRetryEntry 4 }

sipCommonStatsRetryDisconTime  OBJECT-TYPE
    SYNTAX        TimeStamp
    MAX-ACCESS  read-only
    STATUS        current
    DESCRIPTION
       "The value of the sysUpTime object when the counters for the
        retry statistics objects in this row last experienced a
        discontinuity."
    ::= { sipCommonStatsRetryEntry 5 }

--
-- Other Common Statistics
--
sipCommonOtherStatsTable OBJECT-TYPE
    SYNTAX        SEQUENCE OF SipCommonOtherStatsEntry
    MAX-ACCESS  not-accessible
    STATUS        current
    DESCRIPTION
       "This table contains other common statistics supported by each
        SIP entity in this system."
    ::= { sipCommonOtherStats 1 }

sipCommonOtherStatsEntry OBJECT-TYPE
    SYNTAX        SipCommonOtherStatsEntry
    MAX-ACCESS  not-accessible
    STATUS        current
    DESCRIPTION
       "Information on a particular SIP entity's other common
        statistics.
```

        Each row represents those objects for a particular SIP entity
        present in this system. applIndex is used to uniquely identify
        these instances of SIP entities and correlate them through
        the common framework of the NETWORK-SERVICES-MIB (RFC 2788)."
    INDEX { applIndex }
    ::= { sipCommonOtherStatsTable 1 }

SipCommonOtherStatsEntry ::= SEQUENCE {
        sipCommonOtherStatsNumUnsupportedUris   Counter32,
        sipCommonOtherStatsNumUnsupportedMethods Counter32,
        sipCommonOtherStatsOtherwiseDiscardedMsgs Counter32,
        sipCommonOtherStatsDisconTime   TimeStamp
    }

sipCommonOtherStatsNumUnsupportedUris OBJECT-TYPE
    SYNTAX   Counter32
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
       "Number of RequestURIs received with an unsupported scheme.
        A server normally responds to such requests with a 400 Bad
        Request status code.

        Discontinuities in the value of this counter can occur at
        re-initialization of the SIP entity or service. A Management
        Station can detect discontinuities in this counter by
        monitoring the sipCommonOtherStatsDisconTime object in the same
        row."
    ::= { sipCommonOtherStatsEntry 1 }

sipCommonOtherStatsNumUnsupportedMethods OBJECT-TYPE
    SYNTAX   Counter32
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
       "Number of SIP requests received with unsupported methods. A
        server normally responds to such requests with a 501 (Not
        Implemented) or 405 (Method Not Allowed).

        Discontinuities in the value of this counter can occur at
        re-initialization of the SIP entity or service. A Management
        Station can detect discontinuities in this counter by
        monitoring the sipCommonOtherStatsDisconTime object in the same
        row."
    ::= { sipCommonOtherStatsEntry 2 }

sipCommonOtherStatsOtherwiseDiscardedMsgs OBJECT-TYPE
    SYNTAX   Counter32

```
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
      "Number of SIP messages received that, for any number of
       reasons, was discarded without a response.

       Discontinuities in the value of this counter can occur at
       re-initialization of the SIP entity or service.  A Management
       Station can detect discontinuities in this counter by
       monitoring the sipCommonOtherStatsDisconTime object in the same
       row."
    ::= { sipCommonOtherStatsEntry 3 }

sipCommonOtherStatsDisconTime OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
      "The value of the sysUpTime object when the counters for the
       statistics objects in this row last experienced a
       discontinuity."
    ::= { sipCommonOtherStatsEntry 4 }

--
-- Notification related objects
--


--
-- Status code related notification objects.
--
sipCommonStatusCodeNotifTo OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  accessible-for-notify
    STATUS      current
    DESCRIPTION
      "This object contains the value of the To header in the message
       containing the status code that caused the notification.  The
       header name will be part of this object value.  For example,
       'To: Watson '."
    ::= { sipCommonNotifObjects 1 }

sipCommonStatusCodeNotifFrom OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  accessible-for-notify
    STATUS      current
    DESCRIPTION
      "This object contains the value of the From header in the
       message containing the status code that caused the
```

          notification. The header name will be part of this object
          value. For example, 'From: Watson '."
     ::= { sipCommonNotifObjects 2 }

sipCommonStatusCodeNotifCallId OBJECT-TYPE
     SYNTAX   SnmpAdminString
     MAX-ACCESS accessible-for-notify
     STATUS   current
     DESCRIPTION
        "This object contains the value of the Call-ID in the message
         containing the status code that caused the notification. The
         header name will be part of this object value. For example,
         'Call-ID: 5551212@example.com'."
     ::= { sipCommonNotifObjects 3 }

sipCommonStatusCodeNotifCSeq OBJECT-TYPE
     SYNTAX   Unsigned32
     MAX-ACCESS accessible-for-notify
     STATUS   current
     DESCRIPTION
        "This object contains the CSeq value in the message containing
         the status code that caused the notification. The header name
         will be part of this object value. For example, 'CSeq: 1722
         INVITE'."
     ::= { sipCommonNotifObjects 4 }

--
-- General notification related objects.
--
sipCommonNotifApplIndex OBJECT-TYPE
     SYNTAX   Unsigned32 (1..2147483647)
     MAX-ACCESS accessible-for-notify
     STATUS   current
     DESCRIPTION
        "This object contains the applIndex as described in RFC 2788.
         This object is created in order to allow a variable binding
         containing a value of applIndex in a notification."
     ::= { sipCommonNotifObjects 5 }

sipCommonNotifSequenceNumber OBJECT-TYPE
     SYNTAX   Unsigned32 (1..2147483647)
     MAX-ACCESS accessible-for-notify
     STATUS   current
     DESCRIPTION
        "This object contains a sequence number for each notification
         generated by this SIP entity. Each notification SHOULD have a
         unique sequence number. A network manager can use this
         information to determine whether notifications from a

          particular SIP entity have been missed. The value of this
          object MUST start at 1 and increase by 1 with each generated
          notification. If a system restarts, the sequence number MAY
          start again from 1."
     ::= { sipCommonNotifObjects 6 }

--
-- Notifications
--
sipCommonStatusCodeNotif NOTIFICATION-TYPE
     OBJECTS {
         sipCommonNotifSequenceNumber,
         sipCommonNotifApplIndex,
         sipCommonStatusCodeNotifTo,
         sipCommonStatusCodeNotifFrom,
         sipCommonStatusCodeNotifCallId,
         sipCommonStatusCodeNotifCSeq,
         sipCommonStatusCodeIns,
         sipCommonStatusCodeOuts
     }
     STATUS   current
     DESCRIPTION
       "Signifies that a specific status code has been sent or received
        by the system."
     ::= { sipCommonMIBNotifications 1 }

sipCommonStatusCodeThreshExceededInNotif NOTIFICATION-TYPE
     OBJECTS {
         sipCommonNotifSequenceNumber,
         sipCommonNotifApplIndex,
         sipCommonStatusCodeIns
     }
     STATUS   current
     DESCRIPTION
       "Signifies that a specific status code was found to have been
        received by the system frequently enough to exceed the
        configured threshold. This notification can be used as
        an early warning mechanism in lieu of using
        sipCommonStatusCodeNotif."
     ::= { sipCommonMIBNotifications 2 }

sipCommonStatusCodeThreshExceededOutNotif NOTIFICATION-TYPE
     OBJECTS {
         sipCommonNotifSequenceNumber,
         sipCommonNotifApplIndex,
         sipCommonStatusCodeOuts
     }
     STATUS   current

```
    DESCRIPTION
        "Signifies that a specific status code was found to have been
         sent by the system enough to exceed the configured threshold.
         This notification can be used as an early warning mechanism in
         lieu of using sipCommonStatusCodeNotif."
    ::= { sipCommonMIBNotifications 3 }

sipCommonServiceColdStart NOTIFICATION-TYPE
    OBJECTS {
        sipCommonNotifSequenceNumber,
        sipCommonNotifApplIndex,
        sipCommonCfgServiceStartTime
    }
    STATUS       current
    DESCRIPTION
        "Signifies that the SIP service has reinitialized itself or
         started for the first time.  This SHOULD result from a hard
         'down' to 'up' administrative status change.  The configuration
         or behavior of the service MAY be altered."
    ::= { sipCommonMIBNotifications 4 }

sipCommonServiceWarmStart NOTIFICATION-TYPE
    OBJECTS {
        sipCommonNotifSequenceNumber,
        sipCommonNotifApplIndex,
        sipCommonCfgServiceLastChange
    }
    STATUS       current
    DESCRIPTION
        "Signifies that the SIP service has reinitialized itself and is
         restarting after an administrative 'reset'.  The configuration
         or behavior of the service MAY be altered."
    ::= { sipCommonMIBNotifications 5 }

sipCommonServiceStatusChanged NOTIFICATION-TYPE
    OBJECTS {
        sipCommonNotifSequenceNumber,
        sipCommonNotifApplIndex,
        sipCommonCfgServiceLastChange,
        sipCommonCfgServiceOperStatus
    }
    STATUS       current
    DESCRIPTION
        "Signifies that the SIP service operational status has changed."
    ::= { sipCommonMIBNotifications 6 }

--
-- Conformance
```

```
--
sipCommonMIBCompliances
    OBJECT IDENTIFIER ::= { sipCommonMIBConformance 1 }
sipCommonMIBGroups
    OBJECT IDENTIFIER ::= { sipCommonMIBConformance 2 }


--
-- Compliance Statements
--
sipCommonCompliance MODULE-COMPLIANCE
    STATUS      current
    DESCRIPTION
       "The compliance statement for SIP entities."

    MODULE -- this module
        MANDATORY-GROUPS { sipCommonConfigGroup,
                           sipCommonStatsGroup
                         }

    OBJECT        sipCommonStatusCodeRowStatus
    SYNTAX        RowStatus { active(1) }
    WRITE-SYNTAX RowStatus { createAndGo(4), destroy(6) }
    DESCRIPTION
       "Support for createAndWait and notInService is not required."

    OBJECT        sipCommonCfgServiceNotifEnable
    MIN-ACCESS    not-accessible
    DESCRIPTION
       "This object is optional and does not need to be supported."

    GROUP         sipCommonInformationalGroup
    DESCRIPTION
       "This group is OPTIONAL.  A SIP entity can elect to not provide
        any support for these objects, as they provide optional
        information."

    GROUP         sipCommonConfigTimerGroup
    DESCRIPTION
       "This group is OPTIONAL.  A SIP entity can elect to not provide
        any timer configuration."

    GROUP         sipCommonStatsRetryGroup
    DESCRIPTION
       "This group is OPTIONAL.  A SIP entity can elect to not provide
        any retry statistics."

    GROUP         sipCommonNotifGroup
    DESCRIPTION
```

          "This group is OPTIONAL. A SIP entity can elect to not provide
           any notifications. If implemented, the
           sipCommonStatusCodeNotifGroup and sipCommonNotifObjectsGroup
           MUST also be implemented."

      GROUP   sipCommonStatusCodeNotifGroup
      DESCRIPTION
          "This group is OPTIONAL. A SIP entity can elect to not provide
           any notifications. If implemented, the sipCommonNotifGroup and
           sipCommonNotifObjectsGroup MUST also be implemented."

      GROUP   sipCommonNotifObjectsGroup
      DESCRIPTION
          "This group is OPTIONAL. A SIP entity can elect to not provide
           any notifications. If implemented, the
           sipCommonStatusCodeNotifGroup and sipCommonNotifGroup MUST also
           be implemented."

      ::= { sipCommonMIBCompliances 1 }

--
-- Units of Conformance
--
sipCommonConfigGroup OBJECT-GROUP
      OBJECTS {
              sipCommonCfgProtocolVersion,
              sipCommonCfgServiceOperStatus,
              sipCommonCfgServiceStartTime,
              sipCommonCfgServiceLastChange,
              sipCommonPortTransportRcv,
              sipCommonOptionTag,
              sipCommonOptionTagHeaderField,
              sipCommonCfgMaxTransactions,
              sipCommonCfgServiceNotifEnable,
              sipCommonCfgEntityType,
              sipCommonMethodSupportedName
      }
      STATUS current
      DESCRIPTION
         "A collection of objects providing configuration common to all
          SIP entities."
      ::= { sipCommonMIBGroups 1 }

sipCommonInformationalGroup OBJECT-GROUP
      OBJECTS {
              sipCommonCfgOrganization
      }
      STATUS current

```
    DESCRIPTION
        "A collection of objects providing configuration common to all
         SIP entities."
    ::= { sipCommonMIBGroups 2 }

sipCommonConfigTimerGroup OBJECT-GROUP
    OBJECTS {
            sipCommonCfgTimerA,
            sipCommonCfgTimerB,
            sipCommonCfgTimerC,
            sipCommonCfgTimerD,
            sipCommonCfgTimerE,
            sipCommonCfgTimerF,
            sipCommonCfgTimerG,
            sipCommonCfgTimerH,
            sipCommonCfgTimerI,
            sipCommonCfgTimerJ,
            sipCommonCfgTimerK,
            sipCommonCfgTimerT1,
            sipCommonCfgTimerT2,
            sipCommonCfgTimerT4
    }
    STATUS  current
    DESCRIPTION
        "A collection of objects providing timer configuration common to
         all SIP entities."
    ::= { sipCommonMIBGroups 3 }

sipCommonStatsGroup OBJECT-GROUP
    OBJECTS {
            sipCommonSummaryInRequests,
            sipCommonSummaryOutRequests,
            sipCommonSummaryInResponses,
            sipCommonSummaryOutResponses,
            sipCommonSummaryTotalTransactions,
            sipCommonSummaryDisconTime,
            sipCommonMethodStatsOutbounds,
            sipCommonMethodStatsInbounds,
            sipCommonMethodStatsDisconTime,
            sipCommonStatusCodeIns,
            sipCommonStatusCodeOuts,
            sipCommonStatusCodeRowStatus,
            sipCommonStatusCodeDisconTime,
            sipCommonTransCurrentactions,
            sipCommonOtherStatsNumUnsupportedUris,
            sipCommonOtherStatsNumUnsupportedMethods,
            sipCommonOtherStatsOtherwiseDiscardedMsgs,
            sipCommonOtherStatsDisconTime
```

```
    }
    STATUS  current
    DESCRIPTION
      "A collection of objects providing statistics common to all SIP
       entities."
    ::= { sipCommonMIBGroups 4 }

sipCommonStatsRetryGroup OBJECT-GROUP
    OBJECTS {
            sipCommonStatsRetries,
            sipCommonStatsRetryFinalResponses,
            sipCommonStatsRetryNonFinalResponses,
            sipCommonStatsRetryDisconTime
    }
    STATUS  current
    DESCRIPTION
      "A collection of objects providing retry statistics."
    ::= { sipCommonMIBGroups 5 }

sipCommonNotifGroup NOTIFICATION-GROUP
    NOTIFICATIONS {
            sipCommonStatusCodeNotif,
            sipCommonStatusCodeThreshExceededInNotif,
            sipCommonStatusCodeThreshExceededOutNotif,
            sipCommonServiceColdStart,
            sipCommonServiceWarmStart,
            sipCommonServiceStatusChanged
    }

    STATUS  current
    DESCRIPTION
      "A collection of notifications common to all SIP entities."
    ::= { sipCommonMIBGroups 6 }

sipCommonStatusCodeNotifGroup OBJECT-GROUP
    OBJECTS {
            sipCommonStatusCodeNotifSend,
            sipCommonStatusCodeNotifEmitMode,
            sipCommonStatusCodeNotifThresh,
            sipCommonStatusCodeNotifInterval
   }
    STATUS  current
    DESCRIPTION
      "A collection of objects related to the control and attribution
       of notifications common to all SIP entities."
    ::= { sipCommonMIBGroups 7 }

sipCommonNotifObjectsGroup OBJECT-GROUP
```

```
        OBJECTS {
                sipCommonStatusCodeNotifTo,
                sipCommonStatusCodeNotifFrom,
                sipCommonStatusCodeNotifCallId,
                sipCommonStatusCodeNotifCSeq,
                sipCommonNotifApplIndex,
                sipCommonNotifSequenceNumber
        }
        STATUS   current
        DESCRIPTION
          "A collection of accessible-for-notify objects related to the
           notification defined in this MIB module."
        ::= { sipCommonMIBGroups 8 }

END
```

7.3. SIP User Agent MIB Module

```
SIP-UA-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY,
    OBJECT-TYPE,
    Unsigned32,
    mib-2
            FROM SNMPv2-SMI              -- RFC 2578

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

    applIndex
            FROM NETWORK-SERVICES-MIB    -- RFC 2788

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

    SipTCEntityRole
            FROM SIP-TC-MIB;             -- RFC 4780

sipUAMIB MODULE-IDENTITY
    LAST-UPDATED   "200704200000Z"
    ORGANIZATION   "IETF Session Initiation Protocol Working Group"
    CONTACT-INFO
            "SIP WG email: sip@ietf.org

             Co-editor  Kevin Lingle
```

                         Cisco Systems, Inc.
             postal: 7025 Kit Creek Road
                         P.O. Box 14987
                         Research Triangle Park, NC 27709
                         USA
             email: klingle@cisco.com
             phone: +1 919 476 2029

             Co-editor   Joon Maeng
             email: jmaeng@austin.rr.com

             Co-editor   Jean-Francois Mule
                         CableLabs
             postal: 858 Coal Creek Circle
                         Louisville, CO 80027
                         USA
             email: jf.mule@cablelabs.com
             phone: +1 303 661 9100

             Co-editor   Dave Walker
             email: drwalker@rogers.com"
        DESCRIPTION
           "Session Initiation Protocol (SIP) User Agent (UA) MIB module.

            SIP is an application-layer signaling protocol for creating,
            modifying, and terminating multimedia sessions with one or more
            participants. These sessions include Internet multimedia
            conferences and Internet telephone calls. SIP is defined in
            RFC 3261 (June 2002).

            A User Agent is an application that contains both a User Agent
            Client (UAC) and a User Agent Server (UAS). A UAC is an
            application that initiates a SIP request. A UAS is an
            application that contacts the user when a SIP request is
            received and that returns a response on behalf of the user.
            The response accepts, rejects, or redirects the request.

            Copyright (C) The IETF Trust (2007). This version of
            this MIB module is part of RFC 4780; see the RFC itself for
            full legal notices."
        REVISION   "200704200000Z"
        DESCRIPTION
           "Initial version of the IETF SIP-UA-MIB module. This version
            published as part of RFC 4780."
         ::= { mib-2 150 }

-- Top-Level Components of this MIB.
sipUAMIBObjects   OBJECT IDENTIFIER ::= { sipUAMIB 1 }

```
sipUAMIBConformance     OBJECT IDENTIFIER ::= { sipUAMIB 2 }

--
-- This MIB contains objects related to SIP User Agents.
--
sipUACfgServer          OBJECT IDENTIFIER ::= { sipUAMIBObjects 1 }

--
-- SIP Server Configuration
--
sipUACfgServerTable OBJECT-TYPE
    SYNTAX        SEQUENCE OF SipUACfgServerEntry
    MAX-ACCESS  not-accessible
    STATUS        current
    DESCRIPTION
      "This table contains SIP server configuration objects applicable
       to each SIP user agent in this system."
    ::= { sipUACfgServer 1 }

sipUACfgServerEntry OBJECT-TYPE
    SYNTAX        SipUACfgServerEntry
    MAX-ACCESS  not-accessible
    STATUS        current
    DESCRIPTION
      "A row of server configuration.

       Each row represents those objects for a particular SIP user
       agent present in this system.  applIndex is used to uniquely
       identify these instances of SIP user agents and correlate
       them through the common framework of the NETWORK-SERVICES-MIB
       (RFC 2788).  The same value of applIndex used in the
       corresponding SIP-COMMON-MIB is used here."
    INDEX { applIndex, sipUACfgServerIndex }
    ::= { sipUACfgServerTable 1 }

SipUACfgServerEntry ::= SEQUENCE {
        sipUACfgServerIndex       Unsigned32,
        sipUACfgServerAddressType InetAddressType,
        sipUACfgServerAddress     InetAddress,
        sipUACfgServerRole        SipTCEntityRole
    }

sipUACfgServerIndex OBJECT-TYPE
    SYNTAX        Unsigned32 (1..4294967295)
    MAX-ACCESS  not-accessible
    STATUS        current
    DESCRIPTION
      "A unique identifier of a server address when multiple addresses
```

```
         are configured by the SIP entity.  If one address isn't
         reachable, then another can be tried."
      ::= { sipUACfgServerEntry 1 }

sipUACfgServerAddressType OBJECT-TYPE
    SYNTAX        InetAddressType
    MAX-ACCESS  read-only
    STATUS        current
    DESCRIPTION
       "This object reflects the type of address contained in the
        associated instance of sipUACfgServerAddress."
    REFERENCE
       "INET-ADDRESS-MIB (RFC 4001)"
    ::= { sipUACfgServerEntry 2 }

sipUACfgServerAddress OBJECT-TYPE
    SYNTAX        InetAddress
    MAX-ACCESS  read-only
    STATUS        current
    DESCRIPTION
       "This object reflects the address of a SIP server this user
        agent will use to proxy/redirect calls.  The type of this
        address is determined by the value of the
        sipUACfgServerAddressType object."
    REFERENCE "INET-ADDRESS-MIB (RFC 4001)"
    ::= { sipUACfgServerEntry 3 }

sipUACfgServerRole OBJECT-TYPE
    SYNTAX        SipTCEntityRole
    MAX-ACCESS  read-only
    STATUS        current
    DESCRIPTION
       "This object reflects the function of the SIP server this user
        agent should communicate with: registrar, proxy (outbound
        proxy), etc."
    ::= { sipUACfgServerEntry 4 }

--
-- Conformance
--
sipUAMIBCompliances OBJECT IDENTIFIER ::= { sipUAMIBConformance 1 }
sipUAMIBGroups      OBJECT IDENTIFIER ::= { sipUAMIBConformance 2 }

--
-- Compliance Statements
--
sipUACompliance MODULE-COMPLIANCE
    STATUS        current
```

```
      DESCRIPTION
         "The compliance statement for SIP entities that implement the
          SIP-UA-MIB module."
      MODULE -- this module
          MANDATORY-GROUPS { sipUAConfigGroup }

      ::= { sipUAMIBCompliances 1 }

--
-- Units of Conformance
--
sipUAConfigGroup OBJECT-GROUP
    OBJECTS {
            sipUACfgServerAddressType,
            sipUACfgServerAddress,
            sipUACfgServerRole
    }
    STATUS  current
    DESCRIPTION
       "A collection of objects providing information about the
        configuration of SIP User Agents."
    ::= { sipUAMIBGroups 1 }

END
```

7.4. SIP Server MIB Module (Proxy, Redirect, and Registrar Servers)

```
SIP-SERVER-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY,
    OBJECT-TYPE,
    Counter32,
    Unsigned32,
    Gauge32,
    mib-2
        FROM SNMPv2-SMI            -- RFC 2578

    TruthValue,
    TimeStamp, DateAndTime
        FROM SNMPv2-TC            -- RFC 2579

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

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

```
     applIndex
          FROM NETWORK-SERVICES-MIB    -- RFC 2788

     InetAddressType,
     InetAddress
          FROM INET-ADDRESS-MIB;       -- RFC 4001

sipServerMIB MODULE-IDENTITY
     LAST-UPDATED   "200704200000Z"
     ORGANIZATION   "IETF Session Initiation Protocol
                     Working Group"
     CONTACT-INFO
       "SIP WG email: sip@ietf.org

             Co-editor: Kevin Lingle
                        Cisco Systems, Inc.
                postal: 7025 Kit Creek Road
                        P.O. Box 14987
                        Research Triangle Park, NC 27709
                        USA
                 email: klingle@cisco.com
                 phone: +1 919 476 2029

             Co-editor: Joon Maeng
                 email: jmaeng@austin.rr.com

             Co-editor: Jean-Francois Mule
                        CableLabs
                postal: 858 Coal Creek Circle
                        Louisville, CO 80027
                        USA
                 email: jf.mule@cablelabs.com
                 phone: +1 303 661 9100

             Co-editor: Dave Walker
                 email: drwalker@rogers.com
               "
     DESCRIPTION
        "Session Initiation Protocol (SIP) Server MIB module.  SIP is an
         application-layer signaling protocol for creating, modifying,
         and terminating multimedia sessions with one or more
         participants.  These sessions include Internet multimedia
         conferences and Internet telephone calls.  SIP is defined in
         RFC 3261 (June 2002).

         This MIB is defined for the management of SIP Proxy, Redirect,
         and Registrar Servers.
```

          A Proxy Server acts as both a client and a server. It accepts
          requests from other clients, either responding to them or
          passing them on to other servers, possibly after modification.

          A Redirect Server accepts requests from clients and returns
          zero or more addresses to that client. Unlike a User Agent
          Server, it does not accept calls.

          A Registrar is a server that accepts REGISTER requests. A
          Registrar is typically co-located with a Proxy or Redirect
          Server.

          Copyright (C) The IETF Trust (2007). This version of
          this MIB module is part of RFC 4780; see the RFC itself for
          full legal notices."
    REVISION   "200704200000Z"
    DESCRIPTION
        "Initial version of the IETF SIP-SERVER-MIB module. This
        version published as part of RFC 4780."
  ::= { mib-2 151 }

-- Top-Level Components of this MIB.

sipServerMIBObjects   OBJECT IDENTIFIER ::= { sipServerMIB 1 }
sipServerMIBConformance OBJECT IDENTIFIER ::= { sipServerMIB 2 }

--
-- These groups contain objects common to all SIP servers.
--
sipServerCfg   OBJECT IDENTIFIER ::= { sipServerMIBObjects 1 }

--
-- Common Server Configuration Objects
--
sipServerCfgTable OBJECT-TYPE
    SYNTAX   SEQUENCE OF SipServerCfgEntry
    MAX-ACCESS   not-accessible
    STATUS   current
    DESCRIPTION
        "This table contains configuration objects applicable to SIP
        Redirect and Proxy Servers."
    ::= { sipServerCfg 1 }

sipServerCfgEntry OBJECT-TYPE
    SYNTAX   SipServerCfgEntry
    MAX-ACCESS   not-accessible
    STATUS   current
    DESCRIPTION

        "A row of common configuration.

         Each row represents those objects for a particular SIP server
         present in this system. applIndex is used to uniquely identify
         these instances of SIP servers and correlate them through
         the common framework of the NETWORK-SERVICES-MIB (RFC 2788).
         The same value of applIndex used in the corresponding
         SIP-COMMON-MIB is used here."
    INDEX { applIndex }
    ::= { sipServerCfgTable 1 }

SipServerCfgEntry ::=
    SEQUENCE {
        sipServerCfgHostAddressType   InetAddressType,
        sipServerCfgHostAddress   InetAddress
    }

sipServerCfgHostAddressType OBJECT-TYPE
    SYNTAX   InetAddressType
    MAX-ACCESS read-only

    STATUS   current
    DESCRIPTION
        "The type of Internet address by which the SIP server is
         reachable."
    REFERENCE
        "RFC 3261, Section 19.1.1"
    ::= { sipServerCfgEntry 1 }

sipServerCfgHostAddress OBJECT-TYPE
    SYNTAX   InetAddress
    MAX-ACCESS read-only
    STATUS   current
    DESCRIPTION
        "This is the host portion of a SIP URI that is assigned to the
         SIP server. It MAY contain a fully qualified domain name or
         an IP address. The length of the value will depend on the type
         of address specified. The type of address given by this object
         is controlled by sipServerCfgHostAddressType."
    REFERENCE
        "RFC 3261, Section 19.1.1"
    ::= { sipServerCfgEntry 2 }

--
-- This group contains MIB objects
-- related to SIP Proxy Servers.
--
sipServerProxyCfg   OBJECT IDENTIFIER ::= { sipServerMIBObjects 3 }

```
sipServerProxyStats      OBJECT IDENTIFIER ::= { sipServerMIBObjects 4 }

--
-- Proxy Server Configuration
--
sipServerProxyCfgTable OBJECT-TYPE
    SYNTAX        SEQUENCE OF SipServerProxyCfgEntry
    MAX-ACCESS  not-accessible
    STATUS        current
    DESCRIPTION
       "This table contains configuration objects applicable to SIP
        Proxy Servers."
    ::= { sipServerProxyCfg 1 }

sipServerProxyCfgEntry OBJECT-TYPE
    SYNTAX        SipServerProxyCfgEntry
    MAX-ACCESS  not-accessible
    STATUS        current
    DESCRIPTION
       "A row of common proxy configuration.

        Each row represents those objects for a particular SIP server
        present in this system.  applIndex is used to uniquely identify
        these instances of SIP servers and correlate them through the
        common framework of the NETWORK-SERVICES-MIB (RFC 2788).  The
        same value of applIndex used in the corresponding
        SIP-COMMON-MIB is used here."
    INDEX { applIndex }
    ::= { sipServerProxyCfgTable 1 }

SipServerProxyCfgEntry ::=
    SEQUENCE {
        sipServerCfgProxyStatefulness      INTEGER,
        sipServerCfgProxyRecursion         TruthValue,
        sipServerCfgProxyRecordRoute       TruthValue,
        sipServerCfgProxyAuthMethod        BITS,
        sipServerCfgProxyAuthDefaultRealm SnmpAdminString
    }

sipServerCfgProxyStatefulness OBJECT-TYPE
    SYNTAX        INTEGER {
                    stateless(1),
                    transactionStateful(2),
                    callStateful(3)
                  }
    MAX-ACCESS  read-only
    STATUS        current
    DESCRIPTION
```

"This object reflects the default mode of operation for the
Proxy Server entity.

A stateless proxy is a logical entity that does not maintain
the client or server transaction state machines when it
processes requests. A stateless proxy forwards every request it
receives downstream and every response it receives upstream. If
the value of this object is stateless(1), the proxy defaults to
stateless operations.

A transaction stateful proxy, or simply a 'stateful proxy', is
a logical entity that maintains the client and server
transaction state machines during the processing of a request.
A (transaction) stateful proxy is not the same as a call
stateful proxy. If the value of this object is
transactionStateful(2), the proxy is stateful on a transaction
basis.

A call stateful proxy is a logical entity if it retains state
for a dialog from the initiating INVITE to the terminating BYE
request. A call stateful proxy is always transaction stateful,
but the converse is not necessarily true. If the value of this
object is callStateful(3), the proxy is call stateful."
   REFERENCE
       "RFC 3261, Section 16"
   ::= { sipServerProxyCfgEntry 1 }

sipServerCfgProxyRecursion OBJECT-TYPE
    SYNTAX   TruthValue
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
       "This object reflects whether or not the Proxy performs a
        recursive search on the Contacts provided in 3xx redirects.

        If the value of this object is 'true', a recursive search is
        performed. If the value is 'false', no search is performed,
        and the 3xx response is sent upstream towards the source of
        the request."
    REFERENCE
       "RFC 3261 Sections 16.5 and 16.6"
    ::= { sipServerProxyCfgEntry 2 }

sipServerCfgProxyRecordRoute OBJECT-TYPE
    SYNTAX   TruthValue
    MAX-ACCESS read-only
    STATUS   current

        DESCRIPTION
            "This object reflects whether or not the proxy adds itself to
             the Record-Route header as a default action. This header is
             used to list the proxies that insist on being in the signaling
             path for subsequent requests related to the call leg.

             If the value of this object is 'true', the proxy adds itself to
             the end of the Record-Route header, creating the header if
             required. If the value is 'false', the proxy does not add
             itself to the Record-Route header."
        REFERENCE
            "RFC 3261, Section 20.30"
        ::= { sipServerProxyCfgEntry 3 }

--
-- Security
--
sipServerCfgProxyAuthMethod OBJECT-TYPE
    SYNTAX   BITS {
                    none(0),
                    tls(1),
                    digest(2)
                  }
    MAX-ACCESS read-only
    STATUS   current
    DESCRIPTION
        "This object reflects the authentication methods that MAY be
         used to authenticate request originators.

         bit 0 no authentication is performed
         bit 1 TLS is used
         bit 2 HTTP Digest is used."
    REFERENCE
        "RFC 3261 Sections 22, 23, 26, 26.2.3"
    ::= { sipServerProxyCfgEntry 4 }

sipServerCfgProxyAuthDefaultRealm OBJECT-TYPE
    SYNTAX   SnmpAdminString
    MAX-ACCESS read-only
    STATUS   current
    DESCRIPTION
        "This object reflects the default realm value used in
         Proxy-Authenticate headers. Note that this MAY need to be
         stored per user, in which case, this default value is ignored.
        "
    REFERENCE
        "RFC 3261, Section 22.1"
    ::= { sipServerProxyCfgEntry 5 }

```
--
-- Proxy Server Statistics
--
sipServerProxyStatsTable OBJECT-TYPE
    SYNTAX       SEQUENCE OF SipServerProxyStatsEntry
    MAX-ACCESS   not-accessible

    STATUS       current
    DESCRIPTION
       "This table contains the statistics objects applicable to all
        SIP Proxy Servers in this system."
    ::= { sipServerProxyStats 1 }

sipServerProxyStatsEntry OBJECT-TYPE
    SYNTAX       SipServerProxyStatsEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "A row of summary statistics.

        Each row represents those objects for a particular SIP server
        present in this system.  applIndex is used to uniquely identify
        these instances of SIP servers and correlate them through the
        common framework of the NETWORK-SERVICES-MIB (RFC 2788).  The
        same value of applIndex used in the corresponding
        SIP-COMMON-MIB is used here."
    INDEX { applIndex }
    ::= { sipServerProxyStatsTable 1 }

SipServerProxyStatsEntry ::=
    SEQUENCE {
        sipServerProxyStatProxyReqFailures Counter32,
        sipServerProxyStatsDisconTime      TimeStamp
    }

sipServerProxyStatProxyReqFailures OBJECT-TYPE
    SYNTAX       Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "This object contains the number of occurrences of unsupported
        options being specified in received Proxy-Require headers.
        Such occurrences result in a 420 Bad Extension status code
        being returned.

        Discontinuities in the value of this counter can occur at
        re-initialization of the SIP entity or service.  A Management
        Station can detect discontinuities in this counter by
```

            monitoring the sipServerProxyStatsDisconTime object in the same
            row."
    ::= { sipServerProxyStatsEntry 1 }

sipServerProxyStatsDisconTime OBJECT-TYPE
 SYNTAX   TimeStamp
 MAX-ACCESS read-only
 STATUS   current
 DESCRIPTION
    "The value of the sysUpTime object when the counters for the server
     statistics objects in this row last experienced a discontinuity."
 ::= { sipServerProxyStatsEntry 2 }

--
-- This group contains MIB objects related to SIP Registrars.
--
sipServerRegCfg   OBJECT IDENTIFIER ::= { sipServerMIBObjects 5 }
sipServerRegStats   OBJECT IDENTIFIER ::= { sipServerMIBObjects 6 }

--
-- Registrar Configuration
--
sipServerRegCfgTable OBJECT-TYPE
    SYNTAX   SEQUENCE OF SipServerRegCfgEntry
    MAX-ACCESS not-accessible
    STATUS   current
    DESCRIPTION
       "This table contains configuration objects applicable to SIP
        Registrars."
    ::= { sipServerRegCfg 1 }

sipServerRegCfgEntry OBJECT-TYPE
    SYNTAX   SipServerRegCfgEntry
    MAX-ACCESS not-accessible
    STATUS   current
    DESCRIPTION
       "A row of common Registrar configuration.

        Each row represents those objects for a particular SIP server
        present in this system. applIndex is used to uniquely identify
        these instances of SIP servers and correlate them through the
        common framework of the NETWORK-SERVICES-MIB (RFC 2788). The
        same value of applIndex used in the corresponding
        SIP-COMMON-MIB is used here."
    INDEX { applIndex }
    ::= { sipServerRegCfgTable 1 }

SipServerRegCfgEntry ::=

```
    SEQUENCE {
        sipServerRegMaxContactExpiryDuration  Unsigned32,
        sipServerRegMaxUsers                  Unsigned32,
        sipServerRegCurrentUsers              Gauge32,
        sipServerRegDfltRegActiveInterval     Unsigned32
    }

sipServerRegMaxContactExpiryDuration OBJECT-TYPE
    SYNTAX      Unsigned32 (0..4294967295)
    UNITS       "seconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
      "This object reflects the maximum expiry that may be requested
       by a User Agent for a particular Contact.  User Agents can
       specify expiry using either an Expiry header in a REGISTER
       request, or using an Expires parameter in a Contact header in
       a REGISTER request.  If the value requested by the User Agent
       is greater than the value of this object, then the contact
       information is given the duration specified by this object, and
       that duration is indicated to the User Agent in the response."
    ::= { sipServerRegCfgEntry 1 }

sipServerRegMaxUsers OBJECT-TYPE
    SYNTAX      Unsigned32 (1..4294967295)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
      "This object reflects the maximum number of users that the
       Registrar supports.  The current number of users is reflected
       by sipServerRegCurrentUsers."
    ::= { sipServerRegCfgEntry 2 }

sipServerRegCurrentUsers OBJECT-TYPE
    SYNTAX      Gauge32 (0..4294967295)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
      "This object reflects the number of users currently registered
       with the Registrar."
    ::= { sipServerRegCfgEntry 3 }

sipServerRegDfltRegActiveInterval OBJECT-TYPE
    SYNTAX      Unsigned32 (1..4294967295)
    UNITS       "seconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
```

```
            "This object reflects the default time interval the Registrar
             considers registrations to be active.  The value is used to
             compute the Expires header in the REGISTER response.  If a user
             agent requests a time interval shorter than specified by this
             object, the Registrar SHOULD honor that request.  If a Contact
             entry does not have an 'expires' parameter, the value of the
             Expires header field is used instead.  If a Contact entry has no
             'expires' parameter and no Expires header field is present,
             the value of this object is used as the default value."
        REFERENCE
            "RFC 3261, Section 10.2"
        ::= { sipServerRegCfgEntry 4 }
--
-- Per User Information
--
sipServerRegUserTable OBJECT-TYPE
        SYNTAX       SEQUENCE OF SipServerRegUserEntry
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION
            "This table contains information on all users registered to each
             Registrar in this system."
        ::= { sipServerRegCfg 2 }

sipServerRegUserEntry OBJECT-TYPE
        SYNTAX       SipServerRegUserEntry
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION
            "This entry contains information for a single user registered to
             this Registrar.

             Each row represents those objects for a particular SIP server
             present in this system.  applIndex is used to uniquely identify
             these instances of SIP servers and correlate them through the
             common framework of the NETWORK-SERVICES-MIB (RFC 2788).  The
             same value of applIndex used in the corresponding
             SIP-COMMON-MIB is used here."
        INDEX { applIndex, sipServerRegUserIndex }
        ::= { sipServerRegUserTable 1 }

SipServerRegUserEntry ::=
        SEQUENCE {
            sipServerRegUserIndex                 Unsigned32,
            sipServerRegUserUri                   SnmpAdminString,
            sipServerRegUserAuthenticationFailures Counter32,
            sipServerRegUserDisconTime            TimeStamp
        }
```

```
sipServerRegUserIndex OBJECT-TYPE
    SYNTAX        Unsigned32 (1..4294967295)
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "This object uniquely identifies a conceptual row in the table."
    ::= { sipServerRegUserEntry 1 }

sipServerRegUserUri OBJECT-TYPE
    SYNTAX        SnmpAdminString
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "This object contains the user's address-of-record.  It is the
         main form by which the Registrar knows the user.  The format is
         typically 'user@domain'.  It is contained in the To header for
         all REGISTER requests."
    ::= { sipServerRegUserEntry 2 }

sipServerRegUserAuthenticationFailures OBJECT-TYPE
    SYNTAX        Counter32
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "This object contains a count of the number of times the user
         has failed authentication.

         Discontinuities in the value of this counter can occur due to
         successful user authentications and at re-initialization of
         the SIP entity or service.  A Management Station can detect
         discontinuities in this counter by monitoring the
         sipServerRegUserDisconTime object in the same row."
    ::= { sipServerRegUserEntry 3 }

sipServerRegUserDisconTime OBJECT-TYPE
    SYNTAX        TimeStamp
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "The value of the sysUpTime object when the counters for the
         user registration statistics objects in this row last
         experienced a discontinuity."
    ::= { sipServerRegUserEntry 4 }
--
-- Per Contact Information
--
sipServerRegContactTable OBJECT-TYPE
    SYNTAX        SEQUENCE OF SipServerRegContactEntry
```

```
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
      "This table contains information on every location where a
       registered user (specified by sipServerRegUserIndex) wishes to
       be found (i.e., the user has provided contact information to
       each SIP Registrar in this system)."
    ::= { sipServerRegCfg 3 }

sipServerRegContactEntry OBJECT-TYPE
    SYNTAX      SipServerRegContactEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
      "This entry contains information for a single Contact.  Multiple
       contacts may exist for a single user.

       Each row represents those objects for a particular SIP server
       present in this system.  applIndex is used to uniquely identify
       these instances of SIP servers and correlate them through the
       common framework of the NETWORK-SERVICES-MIB (RFC 2788).  The
       same value of applIndex used in the corresponding
       SIP-COMMON-MIB is used here."

    INDEX { applIndex,
            sipServerRegUserIndex,
            sipServerRegContactIndex
          }
    ::= { sipServerRegContactTable 1 }

SipServerRegContactEntry ::=
    SEQUENCE {
        sipServerRegContactIndex        Unsigned32,
        sipServerRegContactDisplayName  SnmpAdminString,
        sipServerRegContactURI          SnmpAdminString,
        sipServerRegContactLastUpdated  TimeStamp,
        sipServerRegContactExpiry       DateAndTime,
        sipServerRegContactPreference   SnmpAdminString
    }

sipServerRegContactIndex OBJECT-TYPE
    SYNTAX      Unsigned32 (1..4294967295)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
      "Along with the sipServerRegUserIndex, this object uniquely
       identifies a conceptual row in the table."
    ::= { sipServerRegContactEntry 1 }
```

sipServerRegContactDisplayName OBJECT-TYPE
    SYNTAX   SnmpAdminString
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
        "This object contains the display name for the Contact. For
         example, 'Santa at Home', or 'Santa on his Sled', corresponding
         to contact URIs of sip:BigGuy@example.com or
         sip:sclaus817@example.com, respectively."
    ::= { sipServerRegContactEntry 2 }

sipServerRegContactURI OBJECT-TYPE
    SYNTAX   SnmpAdminString
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
        "This object contains either a SIP URI where the user can be
         contacted. This URI is normally returned to a client from a
         Redirect Server, or is used as the RequestURI in a SIP request
         line for requests forwarded by a proxy."
    ::= { sipServerRegContactEntry 3 }

sipServerRegContactLastUpdated OBJECT-TYPE
    SYNTAX   TimeStamp
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
        "This object indicates the time when this contact information
         was accepted. If the contact information is updated via a
         subsequent REGISTER of the same information, this object is
         also updated."
    ::= { sipServerRegContactEntry 4 }

sipServerRegContactExpiry OBJECT-TYPE
    SYNTAX   DateAndTime
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
        "This object contains the date and time when the contact
         information will no longer be valid. Such times may be
         specified by the user at registration (i.e., Expires header or
         expiry parameter in the Contact information), or a system
         default can be applied."
    ::= { sipServerRegContactEntry 5 }

sipServerRegContactPreference OBJECT-TYPE
    SYNTAX   SnmpAdminString
    MAX-ACCESS   read-only

```
     STATUS       current
     DESCRIPTION
        "This object indicates a relative preference for the particular
         Contact header field value compared to other bindings for this
         address-of-record.  A registering user may provide this
         preference as a 'qvalue' parameter in the Contact header.

         The format of this item is a decimal number between 0 and 1
         (for example 0.9).  Higher values indicate locations preferred
         by the user."
     REFERENCE
        "RFC 3261, Section 10.2.1.2, 16.6, and 20.10"
     ::= { sipServerRegContactEntry 6 }

--
-- Registrar Statistics
--
sipServerRegStatsTable OBJECT-TYPE
     SYNTAX       SEQUENCE OF SipServerRegStatsEntry
     MAX-ACCESS   not-accessible
     STATUS       current
     DESCRIPTION
        "This table contains the summary statistics objects applicable
         to all SIP Registrars in this system."
     ::= { sipServerRegStats 1 }

sipServerRegStatsEntry OBJECT-TYPE
     SYNTAX       SipServerRegStatsEntry
     MAX-ACCESS   not-accessible
     STATUS       current
     DESCRIPTION
        "A row of summary statistics.

         Each row represents those objects for a particular SIP server
         present in this system.  applIndex is used to uniquely identify
         these instances of SIP servers and correlate them through the
         common framework of the NETWORK-SERVICES-MIB (RFC 2788).  The
         same value of applIndex used in the corresponding
         SIP-COMMON-MIB is used here."
     INDEX { applIndex }
     ::= { sipServerRegStatsTable 1 }

SipServerRegStatsEntry ::=
     SEQUENCE {
         sipServerRegStatsAcceptedRegs      Counter32,
         sipServerRegStatsRejectedRegs      Counter32,
         sipServerRegStatsDisconTime        TimeStamp
     }
```

sipServerRegStatsAcceptedRegs OBJECT-TYPE
    SYNTAX   Counter32
    MAX-ACCESS read-only
    STATUS   current
    DESCRIPTION
        "This object contains a count of the number of REGISTER requests
         that have been accepted (status code 200) by the Registrar.
         This includes additions of new contact information, refreshing
         contact information, as well as requests for deletion of
         contact information.

         Discontinuities in the value of this counter can occur at
         re-initialization of the SIP entity or service. A Management
         Station can detect discontinuities in this counter by
         monitoring the sipServerRegStatsDisconTime object in the same
         row."
    ::= { sipServerRegStatsEntry 1 }

sipServerRegStatsRejectedRegs OBJECT-TYPE
    SYNTAX   Counter32
    MAX-ACCESS read-only
    STATUS   current
    DESCRIPTION
        "This object contains a count of the number REGISTER requests
         that have been rejected by the Registrar.

         Discontinuities in the value of this counter can occur at
         re-initialization of the SIP entity or service. A Management
         Station can detect discontinuities in this counter by
         monitoring the sipServerRegStatsDisconTime object in the same
         row."
  ::= { sipServerRegStatsEntry 2 }

sipServerRegStatsDisconTime OBJECT-TYPE
    SYNTAX   TimeStamp
    MAX-ACCESS read-only
    STATUS   current
    DESCRIPTION
        "The value of the sysUpTime object when the counters for the
         registrar statistics objects in this row last experienced a
         discontinuity."
 ::= { sipServerRegStatsEntry 3 }

--
-- Conformance
--
sipServerMIBCompliances
        OBJECT IDENTIFIER ::= { sipServerMIBConformance 1 }

```
sipServerMIBGroups
        OBJECT IDENTIFIER ::= { sipServerMIBConformance 2 }

--
-- Compliance Statements
--
sipServerProxyServerCompliance MODULE-COMPLIANCE
    STATUS        current
    DESCRIPTION
       "The compliance statement for SIP entities acting as Proxy
        Servers."
    MODULE -- this module
        MANDATORY-GROUPS { sipServerConfigGroup,
                           sipServerProxyConfigGroup,
                           sipServerProxyStatsGroup
                         }
    ::= { sipServerMIBCompliances 1 }

sipRedirectServerCompliance MODULE-COMPLIANCE
    STATUS        current
    DESCRIPTION
       "The compliance statement for SIP entities acting as Redirect
        Servers."
    MODULE -- this module
        MANDATORY-GROUPS { sipServerConfigGroup }
    ::= { sipServerMIBCompliances 2 }

sipServerRegistrarServerCompliance MODULE-COMPLIANCE
    STATUS        current
    DESCRIPTION
       "The compliance statement for SIP entities acting as
        Registrars."
    MODULE -- this module
        MANDATORY-GROUPS { sipServerConfigGroup,
                           sipServerRegistrarConfigGroup,
                           sipServerRegistrarStatsGroup }
    GROUP sipServerRegistrarUsersGroup
    DESCRIPTION
       "This is an optional group."
    ::= { sipServerMIBCompliances 3 }


--
-- Units of Conformance
--
sipServerConfigGroup OBJECT-GROUP
    OBJECTS {
            sipServerCfgHostAddressType,
            sipServerCfgHostAddress
```

```
        }
    STATUS       current
    DESCRIPTION
        "A collection of objects providing configuration common to SIP
         Proxy and Redirect servers."
    ::= { sipServerMIBGroups 1 }

sipServerProxyConfigGroup OBJECT-GROUP
    OBJECTS {
            sipServerCfgProxyStatefulness,
            sipServerCfgProxyRecursion,
            sipServerCfgProxyRecordRoute,
            sipServerCfgProxyAuthMethod,
            sipServerCfgProxyAuthDefaultRealm
        }
    STATUS       current
    DESCRIPTION
        "A collection of objects providing configuration for SIP Proxy
         servers."
    ::= { sipServerMIBGroups 2 }

sipServerProxyStatsGroup OBJECT-GROUP
    OBJECTS {
            sipServerProxyStatProxyReqFailures,
            sipServerProxyStatsDisconTime
        }
    STATUS       current
    DESCRIPTION
        "A collection of objects providing statistics for SIP Proxy
         servers."
    ::= { sipServerMIBGroups 3 }

sipServerRegistrarConfigGroup OBJECT-GROUP
    OBJECTS {
            sipServerRegMaxContactExpiryDuration,
            sipServerRegMaxUsers,
            sipServerRegCurrentUsers,
            sipServerRegDfltRegActiveInterval
        }
    STATUS       current
    DESCRIPTION
        "A collection of objects providing configuration for SIP
         Registrars."
    ::= { sipServerMIBGroups 4 }

sipServerRegistrarStatsGroup OBJECT-GROUP
    OBJECTS {
            sipServerRegStatsAcceptedRegs,
```

```
            sipServerRegStatsRejectedRegs,
            sipServerRegStatsDisconTime
    }
    STATUS        current
    DESCRIPTION
      "A collection of objects providing statistics for SIP
       Registrars."
    ::= { sipServerMIBGroups 5 }

sipServerRegistrarUsersGroup OBJECT-GROUP
    OBJECTS {
            sipServerRegUserUri,
            sipServerRegUserAuthenticationFailures,
            sipServerRegUserDisconTime,
            sipServerRegContactDisplayName,
            sipServerRegContactURI,
            sipServerRegContactLastUpdated,
            sipServerRegContactExpiry,
            sipServerRegContactPreference
    }
    STATUS        current
    DESCRIPTION
      "A collection of objects related to registered users."
    ::= { sipServerMIBGroups 6 }

END
```

8. IANA Considerations

The MIB modules defined in this document use the following IANA-
assigned OBJECT IDENTIFIER values recorded in the SMI Numbers
registry:

```
        +---------------+---------------------------+
        | Descriptor    | OBJECT IDENTIFIER value   |
        +---------------+---------------------------+
        | sipTC         | { mib-2 148 }             |
        | sipCommonMIB  | { mib-2 149 }             |
        | sipUAMIB      | { mib-2 150 }             |
        | sipServerMIB  | { mib-2 151 }             |
        +---------------+---------------------------+
```

9. Security Considerations

   There are a number of management objects defined in the SIP-COMMON-
   MIB MIB module with 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.

   The following read-create object in SIP-COMMON-MIB is used to
   configure the status code statistics that will be monitored by the
   SIP entity:

      sipCommonStatusCodeRowStatus:

      If this object is SET maliciously, it may result in an over-
      allocation of resources in a system for the purpose of
      accumulating and maintaining statistics.

   The following read-write objects in SIP-COMMON-MIB are used to
   configure the behavior of certain SNMP notifications potentially
   generated by a SIP entity:

      sipCommonStatusCodeNotifSend, sipCommonStatusCodeNotifEmitMode,
      sipCommonStatusCodeNotifThresh, sipCommonStatusCodeNotifInterval,
      sipCommonCfgServiceNotifEnable:

      If these objects are SET maliciously, it may result in a system
      and/or network performance impact due to the generation of SNMP
      notifications.

   Some of the readable objects in the MIB modules (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.

   The following object values may contain private or confidential
   customer information like first name, last name, customer
   identification, location, company affiliation, the time the
   information was updated, etc.

      sipServerRegContactDisplayName, sipServerRegContactURI,
      sipServerRegContactLastUpdated and sipCommonCfgOrganization.

The sipCommonCfgTable table contains some objects that may help
attackers gain knowledge about the status and operations of the SIP
service. In particular, the object value of
sipCommonCfgServiceOperStatus may indicate that the SIP entity is in
congested state and may lead attackers to build additional service
attacks to overload the system.

The sipCommonCfgEntityType object indicates the type of SIP entity,
and the sipCommonMethodSupportedTable table contains in the SIP-
COMMON-MIB MIB module list of SIP methods supported by each entity in
the system. Gaining access to this information may allow attackers
to build method-specific attacks or use unsupported methods to create
denial-of-service attack scenarios.

In the SIP-UA-MIB MIB module, the sipUACfgServerTable contains the
address of the SIP servers providing services to the UA, and
obtaining this information may disclose some private or sensitive
information about the SIP service usage.

In the SIP-SERVER-MIB MIB module, the sipServerCfgProxyAuthMethod
object defines the authentication methods supported by the server and
may be used to build specific denial-of-service attackers targeted at
the security mechanisms employed by the SIP entity.

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 set of MIB modules.

It is RECOMMENDED that implementers consider the security features as
provided by the SNMPv3 framework (see RFC 3410 [RFC3410]), 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
responsi when bility 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.

10. Contributor Acknowledgments

11. References

11.1. Normative References

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

[RFC3261]   Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston,
            A., Peterson, J., Sparks, R., Handley, M., and E.
            Schooler, "SIP: Session Initiation Protocol", RFC 3261,
            June 2002.

[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.

[RFC2788]   Freed, N. and S. Kille, "Network Services Monitoring MIB",
            RFC 2788, 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.

[RFC4001]   Daniele, M., Haberman, B., Routhier, S., and J.
            Schoenwaelder, "Textual Conventions for Internet Network
            Addresses", RFC 4001, February 2005.

11.2. Informative References

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

   [RFC3262] Rosenberg, J. and H. Schulzrinne, "Reliability of
              Provisional Responses in Session Initiation Protocol
              (SIP)", RFC 3262, June 2002.

   [RFC4168] Rosenberg, J., Schulzrinne, H., and G. Camarillo, "The
              Stream Control Transmission Protocol (SCTP) as a Transport
              for the Session Initiation Protocol (SIP)", RFC 4168,
              October 2005.

Authors' Addresses

   Kevin Lingle
   Cisco Systems, Inc.
   7025 Kit Creek Road
   P.O. Box 14987
   Research Triangle Park, NC 27709
   US

   Phone: +1 919 476 2029
   EMail: klingle@cisco.com

   Jean-Francois Mule
   CableLabs
   858 Coal Creek Circle
   Louisville, CO 80027
   US

   Phone: +1 303 661 9100
   EMail: jf.mule@cablelabs.com

   Joon Maeng
   5612 Sedona Drive
   Austin, TX 78759
   US

   Phone: +1 512 418 0590
   EMail: jmaeng@austin.rr.com

   Dave Walker

   EMail: drwalker@rogers.com

Full Copyright Statement

Intellectual Property

Acknowledgement