                 Cable Device Management Information Base for
          Data-Over-Cable Service Interface Specification (DOCSIS)
           Compliant Cable Modems and Cable Modem Termination Systems

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 defines a basic set of managed objects for Simple
   Network Management Protocol (SNMP)-based management of Data Over
   Cable Service Interface Specification (DOCSIS)-compliant Cable Modems
   and Cable Modem Termination Systems.

   This memo obsoletes RFC 2669.

Table of Contents

1. The Internet-Standard Management Framework

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

Managed objects are accessed via a virtual information store, termed
the Management Information Base or MIB. MIB objects are generally
accessed through the Simple Network Management Protocol (SNMP).
Objects in the MIB are defined using the mechanisms defined in the
Structure of Management Information (SMI). This memo specifies a MIB
module that is compliant to the SMIv2, which is described in STD 58,
RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580
[RFC2580].

2. Glossary

The terms in this document are derived either from normal cable
system usage, or from the documents associated with the Data-Over-
Cable Service Interface Specification (DOCSIS) process.

2.1. CATV

Originally "Community Antenna Television", now used to refer to any
cable or hybrid fiber and cable system used to deliver video signals
to a community.

2.2. CM or Cable Modem

A CM acts as a "slave" station in a DOCSIS-compliant cable data
system.

2.3. CMTS or Cable Modem Termination System

A generic term covering a cable bridge or cable router in a head-end.
A CMTS acts as the master station in a DOCSIS-compliant cable data
system. It is the only station that transmits downstream, and it
controls the scheduling of upstream transmissions by its associated
CMs.

2.4. DOCSIS or Data-Over-Cable Service Interface Specification

A term referring to the ITU-T Recommendation J.112 [ITU-T_J.112],
Annex B, standard for cable modem systems. [RFI1.0] [RFI1.1]
[RFI2.0]

2.5. Downstream

   The direction from the head-end towards the subscriber.

2.6. Head-End

   The origination point in most cable systems of the subscriber video
   signals. Generally, also the location of the CMTS equipment.

2.7. Media Access Control (MAC) Packet

   A DOCSIS Packet Data Unit.

2.8. RF

   Radio Frequency.

2.9. Simple Network Management Protocol (SNMP)

   Protocol used for network access to Management Information Base (MIB)
   objects. The three most commonly used versions are Version 1
   (SNMPv1), Version 2 (SNMPv2c), and Version 3 (SNMPv3).

2.10. Upstream

   The direction from the subscriber towards the head-end.

3. Introduction

   This MIB module provides a set of objects required for the management
   of DOCSIS-compliant Cable Modems (CM) and Cable Modem Termination
   Systems (CMTS). The specification is derived from the DOCSIS Radio
   Frequency Interface specification [RFI1.0]. Please note that the
   DOCSIS 1.0 standard only required that Cable Modems implement SNMPv1
   and to process Internet Protocol Version 4 (IPv4) customer traffic.
   Design choices in the original version of this MIB module reflected
   those requirements. DOCSIS 1.1 [RFI1.1] and DOCSIS 2.0 [RFI2.0]
   require support for SNMPv3, as well as for SNMPv1 and SNMPv2c, and
   the changes in this MIB module over the previous proposed standard
   version reflect those additional requirements.

   Future versions of DOCSIS, starting with DOCSIS 3.0 [MULPI3.0], are
   expected to require support for the Internet Protocol Version 6
   (IPv6) as both a Customer Premise Equipment (CPE) protocol and one
   supported by the network elements of the DOCSIS CMTS/CM system.

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

3.1. Structure of the MIB

This MIB module is structured into seven components. A component
contains one or more MIB groups related by deprecation or logical
extension.

o The docsDevBaseGroup extends the MIB-II 'system' group of RFC3418
   [RFC3418] with objects needed for cable device system management.
   Related to this group is the docsDevBaseIgmpGroup (enabling
   Internet Group Management Protocol (IGMP) status and control) and
   the docsDevBaseMaxCpeGroup (managing the maximum number of CPEs
   permitted access through the cable modem).

o The docsDevNmAccessGroup and the docsDevNmAccessExtGroup provide a
   minimum level of SNMP access security (see Section 2.7 of
   [OSSI1.0], Section 2 of [OSSI1.1], and Section 5 of [OSSI2.0]).
   With the completion of the SNMP coexistence document, RFC 3584
   [RFC3584], these groups have been deprecated in this version of
   the MIB.

o The docsDevSoftwareGroup, updated by the docsDevSoftwareGroupV2,
   provides information for network-downloadable software upgrades.
   See "Handling of Software Upgrades", below.

o The docsDevServerGroup, updated by the docsDevServerGroupV2,
   provides information about the progress of the interaction between
   the CM or CMTS and various provisioning servers.

o The docsDevEventGroup, updated by the docsDevEventGroupV2,
   provides control and logging for event reporting. With the
   addition of the SNMP Notification MIB, RFC 3413 [RFC3413], and
   Notification Log MIB, RFC 3014 [RFC3014], which cover event
   reporting, the objects in this MIB module have been modified to
   allow for the usage of these RFCs.

o The docsDevFilterGroup configures filters at the link layer and IP
   layer for bridged data traffic. This group has been deprecated in
   this version of the MIB in favor of the docsDevFilterLLCGroup, and
   by groups from the Differentiated Services MIB [RFC3289] --
   specifically, the groups representing the Data Path, Classifier,
   and Actions tables from that MIB.

   o The docsDevCpeGroup, updated by the docsDevInetCpeGroup, provides
      control over which IP addresses may be used by CPEs (e.g., PCs)
      serviced by a given cable modem. This provides anti-spoofing
      control at the point of origin for a large cable modem system.
      This group is separate from docsDevFilter, primarily as this group
      is only implemented on the Cable Modem (CM) and MUST NOT be
      implemented on the Cable Modem Termination System (CMTS).

3.1.1. IMPORTed MIB Modules and REFERENCE Clauses

   This MIB module IMPORTs definitions normatively from the following
   MIB modules, beyond [RFC2578], [RFC2579], and [RFC2580]: INET-
   ADDRESS-MIB [RFC4001], SNMP-FRAMEWORK-MIB [RFC3411], IF-MIB
   [RFC2863], RMON2-MIB [RFC4502], and DIFFSERV-MIB [RFC3289].

   This MIB module also includes DESCRIPTION and REFERENCE clauses that
   normatively refer to [RFC868], [RFC3617], [RFI1.0], [RFI1.1],
   [RFI2.0], [OSSI1.1], and [OSSI2.0].

3.1.2. Persistence Model for Cable Modems

   Most of the tables in this MIB module (e.g., docsDevNmAccessTable,
   docsDevFilterLLCTable) are specified not to let objects persist
   across reboots.

   The expectation (and current operational practice) is that upon
   reboot, these tables are cleared and repopulated from the DOCSIS
   configuration file supplied by the cable operator. This approach
   enables a cable modem to adapt to the current cable operator's
   environment, which in turn enables cable modem portability across
   different cable operators.

   A notable exception to the persistence model is docsDevEventTable,
   since it is useful to maintain a record of events across reboots for
   debugging purposes.

3.1.3. IPv4 Compliance

   Please note that the compliance statements in this version of the MIB
   module require support only for IPv4 addresses. That is because the
   current versions of the DOCSIS protocols (1.0, 1.1, and 2.0) are not
   IPv6 capable. Although support for IPv6 will require changes to the
   DOCSIS protocols, it is expected that the only changes needed to the
   MIB module itself will be the addition of new compliance statements
   that mandate support for IPv6 addresses.

3.2. Management Requirements

3.2.1. Handling of Software Upgrades

   The Cable Modem software upgrade process is documented in [RFI1.0].
   From a network management station, the operator

   o sets docsDevSwServer to the address of the Trivial File Transfer
      Protocol (TFTP) server for software upgrades;

   o sets docsDevSwFilename to the file pathname of the software
      upgrade image; and

   o sets docsDevSwAdminStatus to upgrade-from-mgt.

   Although DOCSIS only specifies the implementation of the TFTP
   protocol [RFC1350] for file transfers, other functional entities
   embedded within the cable device (particularly a PacketCable
   Multimedia Terminal Adapter [MTA-PROV]) specify the optional
   implementation of the Hyper Text Transfer Protocol (HTTP) [RFC1945]
   and [RFC2616] for file transfers. The value of the
   docsDevSwServerTransportProtocol object determines which protocol is
   used for SNMP-initiated software upgrade.

   One reason for the SNMP-initiated upgrade is to allow loading of a
   temporary software image (e.g., special diagnostic software) that
   differs from the software normally used on that device without
   changing the provisioning database.

   Note that software upgrades should not be accepted blindly by the
   cable device. The cable device may refuse an upgrade if

   o the download is incomplete;

   o the file contents are incomplete or damaged; or

   o the software is not intended for that hardware device (this may
      include the case of a feature set that has not been purchased for
      this device).

   A cable device that implements the code verification mechanisms of
   [BPIPLUS] verifies the source and integrity of the downloaded image
   by validating one or more Code Verification Signatures that are
   bundled within the software upgrade.

3.2.2. Events and Notifications

   This MIB module provides control facilities for reporting events
   through syslog [RFC3164], notifications (traps and informs), and
   non-volatile logging. Additional controls allow the agent to use the
   SNMP Notification MIB [RFC3413] and Notification Log MIB [RFC3014]
   for event notification.

   The conventions for event reporting are outside the scope of this
   document. The definition and coding of common DOCSIS notifications
   can be found in [RFC4547].

3.2.3. Notification Throttling

   The CM and CMTS MUST provide support for notification message
   throttling as described below. The network operator can employ
   notification rate throttling or notification limiting by manipulating
   the appropriate MIB variables.

3.2.3.1. Notification Rate Throttling

   Network operators may employ either of two rate control methods. In
   the first method, the device ceases to send notifications when the
   rate exceeds the specified maximum message rate. It resumes sending
   notifications only if reactivated by a network management station
   request.

   In the second method, the device resumes sending notifications when
   the rate falls below the specified maximum message rate.

   The network operator configures the specified maximum message rate by
   setting the measurement interval (in seconds), and the maximum number
   of notifications to be transmitted within the measurement interval.
   The operator can query the operational throttling state (to determine
   whether notifications are enabled or blocked by throttling) of the
   device, as well as query and set the administrative throttling state
   (to manage the rate control method) of the device.

3.2.3.2. Limiting the Notification Rate

   Network operators may wish to limit the number of notifications sent
   by a device over a specified time period. The device ceases to send
   notifications when the number of notifications exceeds the specified
   threshold. It resumes sending notifications only when the
   measurement interval has passed.

   The network operator defines the maximum number of notifications he
   is willing to handle and sets the measurement interval to a large
   number (in hundredths of a second). For this case, the
   administrative throttling state is set to stop at a threshold that is
   the maximum number of notifications.

   See "Techniques for Managing Asynchronously Generated Alerts"
   [RFC1224] for additional technical motivations.

3.3. Protocol Filters

   The Cable Device MIB provides objects for both Link Layer Control
   (LLC) and IP protocol filters. The LLC protocol filter entries can
   be used to limit CM forwarding to a restricted set of network-layer
   protocols (such as IP, Internetwork Packet Exchange (IPX), Network
   Basic Input/Output System (NetBIOS), and Appletalk).

   The IP protocol filter entries can be used to restrict upstream or
   downstream traffic according to source and destination IP addresses,
   transport-layer protocols (such as Transport Control Protocol (TCP),
   User Datagram Protocol (UDP), and Internet Control Message Protocol
   (ICMP)), and source and destination TCP/UDP port numbers.

   In general, a cable modem applies filters (or, more properly,
   classifiers) in an order appropriate to the layering model.
   Specifically, the inbound MAC (or LLC) layer filters are applied
   first, then the "special" filters, then the IP layer inbound filters,
   then the IP layer outbound filters, and then any final LLC outbound
   filters.

```
                     *****************
                     * LLC Filter In *
                     *****************
                             |
                             V
                  *******************
                  * Special Filters *
                  *       |         *
                  *       V         *
                  *  ***********    *
                  *  * IP Spoof *   *
                  *  ***********    *
                  *       |         *
                  *       V         *
                  * ***************  *
                  * * SNMP Access *  *
                  * ***************  *
                  *       |         *
                  *******************
                          |
                          V
                   ****************
                   * IP Filter In *
                   ****************
                          |
                          V
                   *****************
                   * IP Filter Out *
                   *****************
                          |
                          V
                  ******************
                  * LLC Filter Out *
                  ******************
```
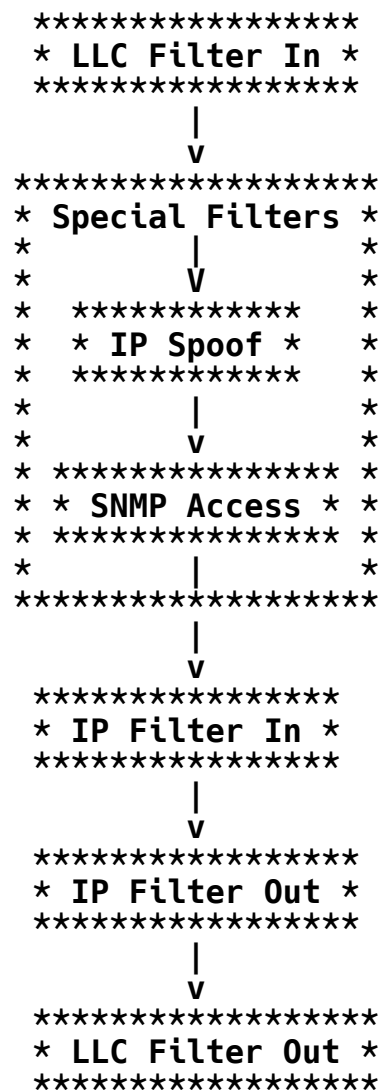
3.3.1. Inbound LLC Filters: docsDevFilterLLCTable

The inbound LLC (or MAC or level-2) filters are contained in the
docsDevFilterLLCTable and are applied to level-2 frames entering the
cable modem from either the RF MAC interface or from one of the CPE
interfaces (physical or logical). These filters are used to prohibit
the processing and forwarding of certain types of level-2 traffic
that may be disruptive to the network. The filters, as currently
specified, can be set to cause the modem either to drop frames that
match at least one filter, or to process a frame that matches at
least one filter. Some examples of possible configurations would be
to permit only IP (and ARP) traffic, or to drop NetBIOS traffic.

3.3.2. Special Filters

   Special filters are applied after the packet is accepted from the MAC
   layer by the IP module, but before any other processing is done.
   They are filters that apply only to a very specific class of traffic.

3.3.2.1. IP Spoofing Filters: docsDevCpeTable, docsDevCpeInetTable

   IP spoofing filters are applied to packets entering the modem from
   one of the CPE interfaces and are intended to prevent a subscriber
   from stealing or misusing IP addresses that were not assigned to the
   subscriber. If the filters are active (enabled), the source address
   of the IP packet must match at least one IP address in one of these
   two tables (docsDevCpeTable or docsDevCpeInetTable), or it is
   discarded without further processing.

   To prevent potential implementation ambiguity, the device consults
   the docsDevCpeTable for the IP packet source address before
   consulting the docsDevCpeInetTable.

   The table can be automatically populated where the first N different
   IP addresses seen from the CPE side of the cable modem are used to
   populate the table automatically. The spoofing filters are specified
   in the docsDevCpeTable and the docsDevCpeInetTable, and the policy
   for automatically creating filters in those tables is controlled by
   docsDevCpeEnroll and docsDevMaxCpe, as well as by the network
   management agent.

   Similar IP spoofing filter controls are defined for CMTS
   implementation in the Subscriber Management MIB [RFC4036].

3.3.2.2. SNMP Access Filters: docsDevNmAccessTable

   The SNMP access filters are applied to SNMP packets entering from any
   interface and destined for the cable modem. If the packets enter
   from a CPE interface, the SNMP filters are applied after the IP
   spoofing filters. The filters only apply to SNMPv1 or SNMPv2c
   traffic and are not consulted for SNMPv3 traffic (and need not be
   implemented by a v3-only agent). SNMPv3 access control is specified
   in the User Security Model MIB, in [RFC3414].

   With the completion of the SNMP coexistence document, RFC 3584
   [RFC3584], docsDevNmAccess table has been deprecated in this version
   of the MIB. See the body of the MIB for the description of how
   agents should handle the interaction between RFC 3584 MIBs and this
   MIB.

3.3.3. IP Filtering: docsDevFilterIpTable

The IP Filtering table acts as a classifier table. Each row in the
table describes a template against which IP packets are compared.
The template includes source and destination addresses (and their
associated masks), upper level protocol (e.g., TCP, UDP), source and
destination port ranges, and Terms of Service (ToS) values. A row
also contains interface and traffic direction match values that have
to be considered in combination. All columns of a particular row
must match the appropriate fields in the packet and must match the
interface and direction items for the packet to result in a match to
the packet.

When classifying a packet, each table is scanned, beginning with the
lowest number filter. If the agent finds a match, it applies the
group of policies specified. If the matched filter has the continue
bit set, the agent continues the scan possibly matching additional
filters and applying additional policies. For example, this allows
the agent to take one set of actions for the 24.0.16/255.255.255.0
group and one set of actions for telnet packets to/from 24.0.16.30,
and these sets of actions may not be mutually exclusive.

Once a packet is matched, one of three actions happen according to
the setting of docsDevFilterIpControl in the row. The packet may be
dropped, in which case no further processing is required. The packet
may be accepted, and processing of the packet continues. Lastly, the
packet may have a set of policy actions applied to it. If
docsDevFilterIpContinue is set to true, scanning of the table
continues and additional matches may result.

When a packet matches and docsDevFilterIpControl in the filter
matched is set to 'policy', the value of docsDevFilterIpPolicyId is
used as a selector into the docsDevFilterPolicyTable. The first
level of indirection may result in zero or more actions being taken
according to the match. The docsDevFilterPolicyTable is scanned in
row order, and all rows where docsDevFilterPolicyId equals
docsDevFilterIpPolicyId have the action specified by the
docsDevFilterPolicyValue 'executed'.

For an example of the use of these IP Filtering MIB tables, see
[RFC2669].

The IP Filtering table and related tables have been deprecated in
this version of the MIB in favor of the Data Path, Classifier, and
Action tables from the Differentiated Services MIB [RFC3289]. See
the body of the MIB for the description of how agents should handle
the interaction between RFC 3289 MIBs and this MIB module.

3.3.4. Outbound LLC Filters

Lastly, any outbound LLC filters are applied to the packet just prior
to its being emitted on the appropriate interface. This MIB module
does not specify any outbound LLC filters, but section 3 of the
DOCSIS Quality of Service (QoS) MIB, [RFC4323], includes outbound LLC
filtering requirements.

4. Definitions

DOCS-CABLE-DEVICE-MIB DEFINITIONS ::= BEGIN

```
IMPORTS
        MODULE-IDENTITY,
        OBJECT-TYPE,
        IpAddress,
        Unsigned32,
        Counter32,
        Integer32,
        zeroDotZero,
        mib-2
                FROM SNMPv2-SMI          -- RFC 2578
        RowStatus,
        RowPointer,
        DateAndTime,
        TruthValue,
        StorageType
                FROM SNMPv2-TC           -- RFC 2579
        InetAddressType,
        InetAddress
                FROM INET-ADDRESS-MIB    -- RFC 4001
        OBJECT-GROUP,
        MODULE-COMPLIANCE
                FROM SNMPv2-CONF         -- RFC 2580
        SnmpAdminString
                FROM SNMP-FRAMEWORK-MIB  -- RFC 3411
        InterfaceIndexOrZero
                FROM IF-MIB              -- RFC 2863
        ZeroBasedCounter32
                FROM RMON2-MIB           -- RFC 4502
        diffServMIBDataPathGroup,
        diffServMIBClfrGroup,
        diffServMIBClfrElementGroup,
        diffServMIBMultiFieldClfrGroup,
        diffServMIBActionGroup,
        diffServMIBDscpMarkActGroup,
        diffServMIBCounterGroup,
        diffServMIBAlgDropGroup,
```

```
                diffServDataPathStatus,
                diffServClfrStatus,
                diffServClfrElementStatus,
                diffServMultiFieldClfrAddrType,
                diffServMultiFieldClfrSrcAddr,
                diffServMultiFieldClfrDstAddr,
                diffServAlgDropStatus,
                diffServDataPathStorage,
                diffServClfrStorage,
                diffServClfrElementStorage,
                diffServMultiFieldClfrStorage,
                diffServActionStorage,
                diffServCountActStorage,
                diffServAlgDropStorage,
                diffServAlgDropType
                        FROM DIFFSERV-MIB;          -- RFC 3289


        docsDev MODULE-IDENTITY
                LAST-UPDATED    "200612200000Z" -- December 20, 2006
                ORGANIZATION    "IETF IP over Cable Data Network
                                 Working Group"
                CONTACT-INFO
                    "        Rich Woundy
                     Postal: Comcast Cable
                             27 Industrial Avenue
                             Chelmsford, MA 01824 U.S.A.
                     Phone:  +1 978 244 4010
                     E-mail: richard_woundy@cable.comcast.com


                             Kevin Marez
                     Postal: Motorola Corporation
                             6450 Sequence Drive
                             San Diego, CA 92121 U.S.A.
                     Phone:  +1 858 404 3785
                     E-mail: kevin.marez@motorola.com

                     IETF IPCDN Working Group
                     General Discussion: ipcdn@ietf.org
                     Subscribe: http://www.ietf.org/mailman/listinfo/ipcdn
                     Archive: ftp://ftp.ietf.org/ietf-mail-archive/ipcdn
                     Co-chairs: Richard Woundy,
                             richard_woundy@cable.comcast.com
                             Jean-Francois Mule,
                             jf.mule@cablelabs.com"

                DESCRIPTION
                    "This is the MIB Module for DOCSIS-compliant cable modems
```

and cable-modem termination systems.

Copyright (C) The IETF Trust (2006). This version
of this MIB module was published in RFC 4639; for full
legal notices see the RFC itself."

REVISION "200612200000Z" -- December 20, 2006
DESCRIPTION
    "Second version, published as RFC 4639.

    Modifications to this MIB module since RFC 2669 include:
    - Deprecation of the docsDevFilter group in favor of the
      DiffServ MIB groups, to enable support for IPv6
      filtering and DiffServ Code Point (DSCP) marking.
    - Deprecation of the docsDevCpeGroup in favor of the
      docsDevCpeInetGroup, to enable support of IPv6.
    - Addition of various InetAddress objects to enable
      support of IPv6.
    - Deprecation of docsDevNmAccessTable in favor of SNMP
      Coexistence and SNMPv3 -- yet adding
      docsDevNmAccessTrapVersion and clarifying
      docsDevNmAccessIp for current use of this table,
    - Addition of docsDevIgmpModeControl for management and
      control of the IGMP mode of operation,
    - Addition of docsDevMaxCpe for management of the
      maxmium number of CPEs permitted access through a
      cable modem,
    - Addition of docsDevSwServerTransportProtocol, and
      modifications to docsDevSoftware object DESCRIPTIONS,
      to enable software downloads via either TFTP or HTTP,
    - Replacement of docsDevEvThrottleInhibited with
      docsDevEvThrottleThresholdExceeded to simplify
      event threshold management,
    - Modification of docsDevEvReporting to enable local
      logging to the internal volatile log, and not to the
      internal non-volatile log,
    - Modification of the compliance statement to make the
      docsDevCpe objects optional
    - Created placeholders for two OIDs in the
      docsDevFilterPolicyTable that were never used
    - Modified the DESCRIPTION of
      docsDevSwServerTransportProtocol and
      docsDevSwServerAddressType to address the
      dependence between each object
    - Added a reference to docsDevServerConfigTftpAddress
    - Clarified the scope of notifications that are covered
      by docsDevEvThrottleThreshold
    - Clarified an error condition that could occur when

doing a SET to docsDevEvReporting
- Defined each of the enumerated types for both
  docsDevEvLevel and docsDevEvPriority
- Added UNITS clause to docsDevFilterLLCMatches,
  docsDevFilterIpMatches, docsDevMaxCpe,
  docsDevEvThrottleThreshold and docsDevEvCounts.
- Added REFERENCE clause to docsDevFilterIpProtocol
- Modified DESCRIPTION of docsDevCpeInetAddr to be
  more protocol-neutral
- Removed the enumerated value (1) from both
  docsDevCpeInetSource and docsDevCpeSource
- Covered additional read-write and read-create objects
  in the Security Considerations section
- Modified the default value of docsDevNmAccessIpMask
  to be consistent with OSSI specification
- Modified the SYNTAX of docsDevNmAccessCommunity and
  docsDevNmAccessInterfaces in the Conformance
  Statement section
- Added SYNTAX clause to docsDevEvReporting in the
  Conformance Statement section
- Modified SYNTAX clause of docsDevEvReporting to
  move new enumerated type to byte boundary
- Added references to DOCSIS 2.0 specifications to
  multiple objects
- Clarified non-persistency across reboots for
  all tables
- Clarified functionality of docsDevSw objects as
  they relate to docsDevSwOperStatus
- Clarified enumerated types (9) and (10) for
  docsDevServerBootState
- Defined the state of unknown(0) for the following
  objects: docsDevServerDhcpAddressType,
  docsDevServerTimeAddressType,
  docsDevServerConfigTftpAddressType and
  docsDevServerConfigTftpAddressType
- Modified the value in docsDevFilterIpDaddr to be
  consistent with the SYNTAX
- Specified which rows could be modified in an
  active row for docsDevFilterPolicyStatus
- Defined the term 'manually' in docsDevCpeEnroll
- Clarified the description for
  docsDevFilterTosOrMask
- Covered the case of a non-existent row for
  docsDevFilterPolicyPtr
- Added DEFVAL clauses for multiple objects
- Replaced docsDevNotification OBJECT IDENTIFIER
  with docsDevNotifications to address possible
  compatibility issues

```
                    - Added support for the usage of RFC 3413 and RFC 3014
                      as event notification mechanisms
                    - Removed docsDevFilterPolicyObsoleteGroup
                    - Added stdInterface(9) type to docsDevEvReporting to
                      support the usage of RFC3413 and RFC3014
                    - Modified DESCRIPTION for docsDevMaxCpe"

            REVISION "199908190000Z"
            DESCRIPTION
                "Initial version, published as RFC 2669."

            ::= { mib-2 69 }

    docsDevMIBObjects  OBJECT IDENTIFIER ::= { docsDev 1 }

    docsDevBase OBJECT IDENTIFIER ::= { docsDevMIBObjects 1 }

    --
    -- For the following object, there is no concept in the
    -- RFI specification corresponding to a backup CMTS.  The
    -- enumeration is provided here in case someone is able
    -- to define such a role or device.
    --

    docsDevRole OBJECT-TYPE
            SYNTAX INTEGER {
                cm(1),
                cmtsActive(2),
                cmtsBackup(3)
            }
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                "Defines the current role of this device.  cm(1) is a
                 Cable Modem, cmtsActive(2) is a Cable Modem Termination
                 System that is controlling the system of cable modems,
                 and cmtsBackup(3) is a CMTS that is currently connected
                 but is not controlling the system (not currently used).

                 In general, if this device is a 'cm', its role will not
                 change during operation or between reboots.  If the
                 device is a 'cmts' it may change between cmtsActive and
                 cmtsBackup and back again during normal operation.  NB:
                 At this time, the DOCSIS standards do not support the
                 concept of a backup CMTS, but cmtsBackup is included for
                 completeness."
            ::= { docsDevBase 1 }
```

```
    docsDevDateTime OBJECT-TYPE
            SYNTAX       DateAndTime
            MAX-ACCESS   read-write
            STATUS       current
            DESCRIPTION
                "The current date and time, with time zone information
                 (if known).

                 If the real data and time cannot be determined, this
                 shall represent elapsed time from boot relative to
                 the standard epoch '1970-1-1,0:0:0.0'.  In other
                 words, if this agent has been up for 3 minutes and
                 not been able to determine what the actual date and
                 time are, this object will return the value
                 '1970-1-1,0:03:0.0'."
            ::= { docsDevBase 2 }

    docsDevResetNow OBJECT-TYPE
            SYNTAX       TruthValue
            MAX-ACCESS   read-write
            STATUS       current
            DESCRIPTION
                "Setting this object to true(1) causes the device to
                 reset.  Reading this object always returns false(2)."
            ::= { docsDevBase 3 }

    docsDevSerialNumber OBJECT-TYPE
            SYNTAX       SnmpAdminString
            MAX-ACCESS   read-only
            STATUS       current
            DESCRIPTION
                "The manufacturer's serial number for this device."
            ::= { docsDevBase 4 }

    docsDevSTPControl OBJECT-TYPE
            SYNTAX INTEGER {
                stEnabled(1),
                noStFilterBpdu(2),
                noStPassBpdu(3)
            }
            MAX-ACCESS   read-write
            STATUS       current
            DESCRIPTION
                "This object controls operation of the spanning tree
                 protocol (as distinguished from transparent bridging).

                 If set to stEnabled(1), then the spanning tree protocol
                 is enabled, subject to bridging constraints.
```

               If noStFilterBpdu(2), then spanning tree is not active,
               and Bridge PDUs received are discarded.

               If noStPassBpdu(3), then spanning tree is not active,
               and Bridge PDUs are transparently forwarded.

               Note that a device need not implement all of these
               options, but that noStFilterBpdu(2) is required."
           DEFVAL { noStFilterBpdu }
           ::= { docsDevBase 5 }

    docsDevIgmpModeControl OBJECT-TYPE
           SYNTAX INTEGER {
               passive(1),
               active(2)
           }
           MAX-ACCESS read-write
           STATUS   current
           DESCRIPTION
               "This object controls the IGMP mode of operation for
               the CM or CMTS. In passive mode, the device forwards
               IGMP between interfaces as based on knowledge of
               Multicast Session activity on the subscriber side
               interface and the rules defined in the DOCSIS RFI
               specification. In active mode, the device terminates
               at and initiates IGMP through its interfaces as based
               on the knowledge of Multicast Session activity on the
               subscriber side interface."
           REFERENCE
               "DOCSIS RFI 1.1 Specification, Section 3.3.1. and
               DOCSIS RFI 2.0 Specification, Section 5.3.1."
           DEFVAL { passive }
           ::= { docsDevBase 6 }

    docsDevMaxCpe OBJECT-TYPE
           SYNTAX   Unsigned32 (0..255)
           UNITS   "CPEs"
           MAX-ACCESS read-only
           STATUS   current
           DESCRIPTION
               "The maximum number of CPEs that can be granted access
               through a CM during a CM epoch. This value can be
               obtained from the CM configuration file; however,
               it may be adjusted by the CM according to hardware or
               software limitations that have been imposed on the
               implementation."
           REFERENCE
               "DOCSIS RFI 1.0 Specification, Appendix C.7.20., and

                    DOCSIS RFI 1.1 Specification, Appendix C.1.1.7. and
                    DOCSIS RFI 2.0 Specification, Appendix C.1.1.7."
             ::= { docsDevBase 7 }

   --
   -- The following table provides one level of security for access
   -- to the device by network management stations.
   -- Note that access is also constrained by the
   -- community strings and any vendor-specific security.
   --

   docsDevNmAccessTable OBJECT-TYPE
           SYNTAX   SEQUENCE OF DocsDevNmAccessEntry
           MAX-ACCESS   not-accessible
           STATUS   deprecated
           DESCRIPTION
               "This table controls access to SNMP objects by network
                management stations. If the table is empty, access to
                SNMP objects is unrestricted. The objects in this table
                MUST NOT persist across reboots. The objects in this
                table are only accessible from cable devices that are
                not capable of operating in SNMP Coexistence mode
                (RFC 3584) or in SNMPv3 mode (RFC 3410).
                See the conformance section for
                details. Note that some devices are required by other
                specifications (e.g., the DOCSIS OSSIv1.1 specification)
                to support the legacy SNMPv1/v2c docsDevNmAccess mode
                for backward compatibility.

                This table is deprecated. Instead, use the SNMP
                coexistence MIBs from RFC 3584, the TARGET and
                NOTIFICATION MIBs from RFC 3413, and
                the View-Based Access Control Model (VACM) MIBs for
                all SNMP protocol versions from RFC 3415."
             ::= { docsDevMIBObjects 2 }

   docsDevNmAccessEntry OBJECT-TYPE
           SYNTAX   DocsDevNmAccessEntry
           MAX-ACCESS   not-accessible
           STATUS   deprecated
           DESCRIPTION
               "An entry describing access to SNMP objects by a
                particular network management station. An entry in
                this table is not readable unless the management station
                has read-write permission (either implicit if the table
                is empty, or explicit through an entry in this table).
                Entries are ordered by docsDevNmAccessIndex. The first

                    matching entry (e.g., matching IP address and community
                    string) is used to derive access."
             INDEX { docsDevNmAccessIndex }
             ::= { docsDevNmAccessTable 1 }

      DocsDevNmAccessEntry ::= SEQUENCE {
                docsDevNmAccessIndex   Integer32,
                docsDevNmAccessIp   IpAddress,
                docsDevNmAccessIpMask   IpAddress,
                docsDevNmAccessCommunity   OCTET STRING,
                docsDevNmAccessControl   INTEGER,
                docsDevNmAccessInterfaces   OCTET STRING,
                docsDevNmAccessStatus   RowStatus,
                docsDevNmAccessTrapVersion   INTEGER
             }

      docsDevNmAccessIndex OBJECT-TYPE
             SYNTAX   Integer32 (1..2147483647)
             MAX-ACCESS not-accessible
             STATUS   deprecated
             DESCRIPTION
                "Index used to order the application of access
                 entries."
             ::= { docsDevNmAccessEntry 1 }

      docsDevNmAccessIp OBJECT-TYPE
             SYNTAX   IpAddress
             MAX-ACCESS read-create
             STATUS   deprecated
             DESCRIPTION
                "The IP address (or subnet) of the network management
                 station. The address 0.0.0.0 is defined to mean
                 any Network Management Station (NMS). If traps are
                 enabled for this entry, then the value must be the
                 address of a specific device. Implementations MAY
                 recognize 255.255.255.255 as equivalent to 0.0.0.0."
             DEFVAL { '00000000'h }
             ::= { docsDevNmAccessEntry 2 }

      docsDevNmAccessIpMask OBJECT-TYPE
             SYNTAX   IpAddress
             MAX-ACCESS read-create
             STATUS   deprecated
             DESCRIPTION
                "The IP subnet mask of the network management stations.
                 If traps are enabled for this entry, then the value must
                 be 0.0.0.0. Implementations MAY recognize
                 255.255.255.255 as equivalent to 0.0.0.0."

```
        DEFVAL { '00000000'h }
        ::= { docsDevNmAccessEntry 3 }

docsDevNmAccessCommunity OBJECT-TYPE
        SYNTAX       OCTET STRING
        MAX-ACCESS   read-create
        STATUS       deprecated
        DESCRIPTION
            "The community string to be matched for access by this
             entry.  If set to a zero-length string, then any
             community string will match.  When read, this object
             SHOULD return a zero-length string."
        DEFVAL { "public" }
        ::= { docsDevNmAccessEntry 4 }

docsDevNmAccessControl OBJECT-TYPE
        SYNTAX           INTEGER {
            none(1),
            read(2),
            readWrite(3),
            roWithTraps(4),
            rwWithTraps(5),
            trapsOnly(6)
        }
        MAX-ACCESS   read-create
        STATUS       deprecated
        DESCRIPTION
            "Specifies the type of access allowed to this NMS.
             Setting this object to none(1) causes the table entry
             to be destroyed.  Read(2) allows access by 'get' and
             'get-next' PDUs.  ReadWrite(3) allows access by 'set' as
             well.  RoWithtraps(4), rwWithTraps(5), and trapsOnly(6)
             control distribution of Trap PDUs transmitted by this
             device."
        DEFVAL { read }
        ::= { docsDevNmAccessEntry 5 }

   -- The syntax of the following object was copied from RFC 1493,
   -- dot1dStaticAllowedToGoTo.

docsDevNmAccessInterfaces OBJECT-TYPE
        SYNTAX       OCTET STRING (SIZE (1..32))
        MAX-ACCESS   read-create
        STATUS       deprecated
        DESCRIPTION
            "Specifies the set of interfaces from which requests from
             this NMS will be accepted.  Each octet within
             the value of this object specifies a set of eight
```

            interfaces, the first octet specifying ports 1
            through 8, the second octet specifying interfaces 9
            through 16, etc. Within each octet, the most
            significant bit represents the lowest numbered
            interface, and the least significant bit represents the
            highest numbered interface. Thus, each interface is
            represented by a single bit within the value of this
            object. If that bit has a value of '1' then that
            interface is included in the set.

            Note that entries in this table apply only to link-layer
            interfaces (e.g., Ethernet and CATV MAC). Bits
            representing upstream and downstream channel interfaces
            MUST NOT be set to '1'.

            Note that if bits corresponding to non-existing
            interfaces are set, the result is implementation
            specific.

            Note that according to the DOCSIS OSSIv1.1
            specification, when ifIndex '1' is included in the
            set, then this row applies to all CPE
            (customer-facing) interfaces.

            The size of this object is the minimum required to
            represent all configured interfaces for this device."
        ::= { docsDevNmAccessEntry 6 }

    docsDevNmAccessStatus OBJECT-TYPE
            SYNTAX   RowStatus
            MAX-ACCESS read-create
            STATUS   deprecated
            DESCRIPTION
            "Controls and reflects the status of rows in this
             table. Rows in this table may be created by either the
             create-and-go or create-and-wait paradigm. There is no
             restriction on changing values in a row of this table
             while the row is active.

             The following objects MUST have valid values before this
             object can be set to active: docsDevNmAccessIp,
             docsDevNmAccessStatus, docsDevNmAccessIpMask,
             docsDevNmAccessCommunity, docsDevNmAccessControl, and
             docsDevNmAccessInterfaces."
        ::= { docsDevNmAccessEntry 7 }

    docsDevNmAccessTrapVersion OBJECT-TYPE
            SYNTAX   INTEGER {

```
                    disableSNMPv2trap(1),
                    enableSNMPv2trap(2)
                }
            MAX-ACCESS  read-create
            STATUS      deprecated
            DESCRIPTION
                "Specifies the TRAP version that is sent to this NMS.
                 Setting this object to disableSNMPv2trap (1) causes the
                 trap in SNMPv1 format to be sent to a particular NMS.
                 Setting this object to enableSNMPv2trap (2) causes the
                 trap in SNMPv2 format be sent to a particular NMS."
            DEFVAL { disableSNMPv2trap }
            ::= { docsDevNmAccessEntry 8 }

    --
    -- The following group describes control objects used for downloading
    -- firmware to a cable device.  Procedures for software download are
    -- described in Section 3.2.1 of the RFC containing this MIB module.
    --

    docsDevSoftware OBJECT IDENTIFIER ::= { docsDevMIBObjects 3 }

    docsDevSwServer OBJECT-TYPE
            SYNTAX      IpAddress
            MAX-ACCESS  read-write
            STATUS      deprecated
            DESCRIPTION
                "The address of the TFTP server used for software
                 upgrades.  If the TFTP server is unknown or is a
                 non-IPv4 address, return 0.0.0.0.

                 This object is deprecated.  See docsDevSwServerAddress
                 for its replacement.  This object will have its value
                 modified, given a valid SET to docsDevSwServerAddress."
            ::= { docsDevSoftware 1 }

    docsDevSwFilename OBJECT-TYPE
            SYNTAX      SnmpAdminString (SIZE (0..64))
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                "The filename of the software image to be downloaded via
                 TFTP, or the abs_path (as defined in RFC 2616) of the
                 software image to be downloaded via HTTP.

                 Unless set via SNMP, this is the filename or abs_path
                 specified by the provisioning server during the boot
                 process that corresponds to the software version that
```

                  is desired for this device.

                  If unknown, the value of this object is the zero-length
                  string."
            ::= { docsDevSoftware 2 }

   docsDevSwAdminStatus OBJECT-TYPE
            SYNTAX INTEGER {
                upgradeFromMgt(1),
                allowProvisioningUpgrade(2),
                ignoreProvisioningUpgrade(3)
            }
            MAX-ACCESS read-write
            STATUS   current
            DESCRIPTION
                "If set to upgradeFromMgt(1), the device will initiate a
                TFTP or HTTP software image download. After
                successfully receiving an image, the device will set
                its state to ignoreProvisioningUpgrade(3) and reboot.
                If the download process is interrupted (e.g., by a reset
                or power failure), the device will load the previous
                image and, after re-initialization, continue to attempt
                loading the image specified in docsDevSwFilename.

                If set to allowProvisioningUpgrade(2), the device will
                use the software version information supplied by the
                provisioning server when next rebooting (this does not
                cause a reboot).

                When set to ignoreProvisioningUpgrade(3), the device
                will disregard software image upgrade information
                from the provisioning server.

                Note that reading this object can return
                upgradeFromMgt(1). This indicates that a software
                download is currently in progress, and that the device
                will reboot after successfully receiving an image."
            DEFVAL { allowProvisioningUpgrade }
            ::= { docsDevSoftware 3 }

   docsDevSwOperStatus OBJECT-TYPE
            SYNTAX INTEGER {
                inProgress(1),
                completeFromProvisioning(2),
                completeFromMgt(3),
                failed(4),
                other(5)
            }

                MAX-ACCESS read-only
                STATUS   current
                DESCRIPTION
                    "InProgress(1) indicates that a TFTP or HTTP download is
                     underway, either as a result of a version mismatch at
                     provisioning or as a result of a upgradeFromMgt request.
                     No other docsDevSw* objects can be modified in
                     this state.

                     CompleteFromProvisioning(2) indicates that the last
                     software upgrade was a result of version mismatch at
                     provisioning.

                     CompleteFromMgt(3) indicates that the last software
                     upgrade was a result of setting docsDevSwAdminStatus to
                     upgradeFromMgt.

                     Failed(4) indicates that the last attempted download
                     failed, ordinarily due to TFTP or HTTP timeout."
                REFERENCE
                    "DOCSIS RFI 1.0 Specification, Section 8.2., and
                     DOCSIS RFI 1.1 Specification, Section 10.1. and
                     DOCSIS RFI 2.0 Specification, Section 12.1."
                ::= { docsDevSoftware 4 }

        docsDevSwCurrentVers OBJECT-TYPE
                SYNTAX   SnmpAdminString
                MAX-ACCESS read-only
                STATUS   current
                DESCRIPTION
                    "The software version currently operating in this device.
                     This string's syntax is that used by the
                     individual vendor to identify software versions.
                     For a CM, this string will describe the current
                     software load. For a CMTS, this object SHOULD contain
                     a human-readable representation either of the vendor
                     specific designation of the software for the chassis,
                     or of the software for the control processor. If
                     neither of these is applicable, the value MUST be a
                     zero-length string."
                ::= { docsDevSoftware 5 }

        docsDevSwServerAddressType OBJECT-TYPE
                SYNTAX   InetAddressType
                MAX-ACCESS read-write
                STATUS   current
                DESCRIPTION
                    "The type of address of the TFTP or HTTP server used for

                    software upgrades.

                    If docsDevSwServerTransportProtocol is currently set to
                    tftp(1), attempting to set this object to dns(16) MUST
                    result in an error."
                ::= { docsDevSoftware 6 }

        docsDevSwServerAddress OBJECT-TYPE
                SYNTAX   InetAddress
                MAX-ACCESS read-write
                STATUS   current
                DESCRIPTION
                    "The address of the TFTP or HTTP server used for software
                    upgrades.

                    If the TFTP/HTTP server is unknown, return the zero-
                    length address string (see the TextualConvention).

                    If docsDevSwServer is also implemented in this agent,
                    this object is tied to it. A set of this object to an
                    IPv4 address will result in also setting the value of
                    docsDevSwServer to that address. If this object is set
                    to an IPv6 address, docsDevSwServer is set to 0.0.0.0.
                    If docsDevSwServer is set, this object is also set to
                    that value. Note that if both are set in the same
                    action, the order of which one sets the other is
                    undefined."
                ::= { docsDevSoftware 7 }

        docsDevSwServerTransportProtocol OBJECT-TYPE
                SYNTAX INTEGER {
                    tftp(1),
                    http(2)
                }
                MAX-ACCESS read-write
                STATUS   current
                DESCRIPTION
                    "This object specifies the transport protocol (TFTP or
                    HTTP) to be used for software upgrades.

                    If the value of this object is tftp(1), then the cable
                    device uses TFTP (RFC 1350) read request packets to
                    download the docsDevSwFilename from the
                    docsDevSwServerAddress in octet mode.

                    If the value of this object is http(2), then the cable
                    device uses HTTP 1.0 (RFC 1945) or HTTP 1.1 (RFC 2616)
                    GET requests sent to host docsDevSwServerAddress to

                download the software image from path docsDevSwFilename.

                If docsDevSwServerAddressType is currently set to
                dns(16), attempting to set this object to tftp(1) MUST
                result in an error."
            DEFVAL { tftp }
            ::= { docsDevSoftware 8 }

    --
    -- The following group describes server access and parameters used
    -- for initial provisioning and bootstrapping.
    --

    docsDevServer OBJECT IDENTIFIER ::= { docsDevMIBObjects 4 }

    docsDevServerBootState OBJECT-TYPE
            SYNTAX INTEGER {
                operational(1),
                disabled(2),
                waitingForDhcpOffer(3),
                waitingForDhcpResponse(4),
                waitingForTimeServer(5),
                waitingForTftp(6),
                refusedByCmts(7),
                forwardingDenied(8),
                other(9),
                unknown(10)
            }
            MAX-ACCESS   read-only
            STATUS   current
            DESCRIPTION
                "If operational(1), the device has completed loading and
                 processing of configuration parameters, and the CMTS has
                 completed the Registration exchange.

                 If disabled(2), then the device was administratively
                 disabled, possibly by being refused network access in
                 the configuration file.

                 If waitingForDhcpOffer(3), then a Dynamic Host
                 Configuration Protocol (DHCP) Discover has been
                 transmitted, and no offer has yet been received.

                 If waitingForDhcpResponse(4), then a DHCP Request has
                 been transmitted, and no response has yet been received.

                 If waitingForTimeServer(5), then a Time Request has been
                 transmitted, and no response has yet been received.

                  If waitingForTftp(6), then a request to the TFTP
                  parameter server has been made, and no response
                  received.

                  If refusedByCmts(7), then the Registration
                  Request/Response exchange with the CMTS failed.

                  If forwardingDenied(8), then the registration process
                  was completed, but the network access option in the
                  received configuration file prohibits forwarding.

                  If other(9), then the registration process reached a
                  point that does not fall into one of the above
                  categories.

                  If unknown(10), then the device has not yet begun the
                  registration process or is in some other indeterminate
                  state."
             REFERENCE
                 "DOCSIS RFI 1.0 Specification, Figure 7-1, and
                  DOCSIS RFI 1.1 Specification, Figure 9-1 and
                  DOCSIS RFI 2.0 Specification, Figure 11-1."
             ::= { docsDevServer 1 }

      docsDevServerDhcp OBJECT-TYPE
             SYNTAX   IpAddress
             MAX-ACCESS   read-only
             STATUS   deprecated
             DESCRIPTION
                 "The IP address of the DHCP server that assigned an IP
                  address to this device. Returns 0.0.0.0 if DHCP is not
                  used for IP address assignment, or if this agent is
                  not assigned an IPv4 address.

                  This object is deprecated and is replaced by
                  docsDevServerDhcpAddress."
             ::= { docsDevServer 2 }

      docsDevServerTime OBJECT-TYPE
             SYNTAX   IpAddress
             MAX-ACCESS   read-only
             STATUS   deprecated
             DESCRIPTION
                 "The IP address of the Time server (RFC 0868). Returns
                  0.0.0.0 if the time server IP address is unknown, or if
                  the time server is not an IPv4 server.

                  This object is deprecated and is replaced by

```
                    docsDevServerTimeAddress."
            ::= { docsDevServer 3 }

    docsDevServerTftp OBJECT-TYPE
            SYNTAX       IpAddress
            MAX-ACCESS   read-only
            STATUS       deprecated
            DESCRIPTION
                "The IP address of the TFTP server responsible for
                 downloading provisioning and configuration parameters
                 to this device.  Returns 0.0.0.0 if the TFTP server
                 address is unknown or is not an IPv4 address.

                 This object is deprecated and is replaced by
                 docsDevServerConfigTftpAddress."
            ::= { docsDevServer 4 }

    docsDevServerConfigFile OBJECT-TYPE
            SYNTAX       SnmpAdminString
            MAX-ACCESS   read-only
            STATUS       current
            DESCRIPTION
                "The name of the device configuration file read from
                 the TFTP server.  Returns a zero-length string if
                 the configuration file name is unknown."
            ::= { docsDevServer 5 }

    docsDevServerDhcpAddressType OBJECT-TYPE
            SYNTAX       InetAddressType
            MAX-ACCESS   read-only
            STATUS       current
            DESCRIPTION
                "The type of address of docsDevServerDhcpAddress.  If
                 DHCP was not used, this value should return
                 unknown(0)."
            ::= { docsDevServer 6 }

    docsDevServerDhcpAddress OBJECT-TYPE
            SYNTAX       InetAddress
            MAX-ACCESS   read-only
            STATUS       current
            DESCRIPTION
                "The internet address of the DHCP server that assigned
                 an IP address to this device.  Returns the zero length
                 octet string if DHCP was not used for IP address
                 assignment."
            ::= { docsDevServer 7 }
```

       docsDevServerTimeAddressType OBJECT-TYPE
               SYNTAX   InetAddressType
               MAX-ACCESS   read-only
               STATUS   current
               DESCRIPTION
                   "The type of address of docsDevServerTimeAddress. If
                    no time server exists, this value should return
                    unknown(0)."
               ::= { docsDevServer 8 }

       docsDevServerTimeAddress OBJECT-TYPE
               SYNTAX   InetAddress
               MAX-ACCESS   read-only
               STATUS   current
               DESCRIPTION
                   "The Internet address of the RFC 868 Time server,
                    as provided by DHCP option 4.

                    Note that if multiple values are provided to the
                    CM in DHCP option 4, the value of this MIB object
                    MUST be the Time server address from which the Time
                    of Day reference was acquired as based on the DOCSIS
                    RFI specification. During the period of time where
                    the Time of Day have not been acquired, the Time
                    server address reported by the CM may report the
                    first address value in the DHCP option value or the
                    last server address the CM attempted to get the Time
                    of day value.

                    Returns the zero-length octet string if the time server
                    IP address is not provisioned."
               REFERENCE
                   "DOCSIS RFI 1.1 Specification, Section 9.2.7. and
                    DOCSIS RFI 2.0 Specification, Section 11.2.7."
               ::= { docsDevServer 9 }

     docsDevServerConfigTftpAddressType OBJECT-TYPE
               SYNTAX   InetAddressType
               MAX-ACCESS   read-only
               STATUS   current
               DESCRIPTION
                   "The type of address of docsDevServerConfigTftpAddress.
                    If no TFTP server exists, this value should return
                    unknown(0)."
               ::= { docsDevServer 10 }

     docsDevServerConfigTftpAddress OBJECT-TYPE
               SYNTAX   InetAddress

```
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The internet address of the TFTP server responsible for
             downloading provisioning and configuration parameters
             to this device.  Returns the zero-length octet string if
             the config server address is unknown.  There are certain
             security risks that are involved with using TFTP."
        REFERENCE
            "RFC 3617, Section 5"
        ::= { docsDevServer 11 }


--
-- Event Reporting
--

docsDevEvent OBJECT IDENTIFIER ::= { docsDevMIBObjects 5 }

docsDevEvControl OBJECT-TYPE
        SYNTAX INTEGER {
            resetLog(1),
            useDefaultReporting(2)
        }
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "Setting this object to resetLog(1) empties the event
             log.  All data is deleted.  Setting it to
             useDefaultReporting(2) returns all event priorities to
             their factory-default reporting.  Reading this object
             always returns useDefaultReporting(2)."
        ::= { docsDevEvent 1 }

docsDevEvSyslog OBJECT-TYPE
        SYNTAX      IpAddress
        MAX-ACCESS  read-write
        STATUS      deprecated
        DESCRIPTION
            "The IP address of the Syslog server.  If 0.0.0.0, either
             syslog transmission is inhibited, or the Syslog server
             address is not an IPv4 address.

             This object is deprecated and is replaced by
             docsDevEvSyslogAddress."
        ::= { docsDevEvent 2 }

docsDevEvThrottleAdminStatus OBJECT-TYPE
```

```
            SYNTAX INTEGER {
                unconstrained(1),
                maintainBelowThreshold(2),
                stopAtThreshold(3),
                inhibited(4)
            }
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                "Controls the transmission of traps and syslog messages
                 with respect to the trap pacing threshold.

                 unconstrained(1) causes traps and syslog messages to be
                 transmitted without regard to the threshold settings.

                 maintainBelowThreshold(2) causes trap transmission and
                 syslog messages to be suppressed if the number of traps
                 would otherwise exceed the threshold.

                 stopAtThreshold(3) causes trap transmission to cease at
                 the threshold and not to resume until directed to do so.

                 inhibited(4) causes all trap transmission and syslog
                 messages to be suppressed.

                 A single event is always treated as a single event for
                 threshold counting.  That is, an event causing both a
                 trap and a syslog message is still treated as a single
                 event.

                 Writing to this object resets the thresholding state."
            DEFVAL { unconstrained }
            ::= { docsDevEvent 3 }

    docsDevEvThrottleInhibited OBJECT-TYPE
            SYNTAX      TruthValue
            MAX-ACCESS  read-only
            STATUS      deprecated
            DESCRIPTION
                "If true(1), trap and syslog transmission is currently
                 inhibited due to thresholds and/or the current setting
                 of docsDevEvThrottleAdminStatus.  In addition, this is
                 true(1) when transmission is inhibited because no
                 syslog (docsDevEvSyslog) or trap (docsDevNmAccessEntry)
                 destinations have been set.

                 This object is deprecated and is replaced by
                 docsDevEvThrottleThresholdExceeded."
```

```
            ::= { docsDevEvent 4 }

    docsDevEvThrottleThreshold OBJECT-TYPE
            SYNTAX       Unsigned32
            UNITS        "events"
            MAX-ACCESS   read-write
            STATUS       current
            DESCRIPTION
                "Number of events per docsDevEvThrottleInterval permitted
                 before throttling is to occur.

                 A single event, whether the notification could result in
                 messages transmitted using syslog, SNMP, or both
                 protocols, and regardless of the number of destinations,
                 (including zero) is always treated as a single event for
                 threshold counting.  For example, an event causing both
                 a trap and a syslog message is still treated as a single
                 event.

                 All system notifications that occur within the device
                 should be taken into consideration when calculating
                 and monitoring the threshold."
            DEFVAL { 0 }
            ::= { docsDevEvent 5 }

    docsDevEvThrottleInterval OBJECT-TYPE
            SYNTAX       Integer32 (1..2147483647)
            UNITS        "seconds"
            MAX-ACCESS   read-write
            STATUS       current
            DESCRIPTION
                "The interval over which docsDevEvThrottleThreshold
                 applies."
            DEFVAL { 1 }
            ::= { docsDevEvent 6 }

    --
    -- The following table controls the reporting of the various classes
    -- of events.
    --

    docsDevEvControlTable OBJECT-TYPE
            SYNTAX       SEQUENCE OF DocsDevEvControlEntry
            MAX-ACCESS   not-accessible
            STATUS       current
            DESCRIPTION
                "This table allows control of the reporting of event
                 classes.  For each event priority, a combination of
```

                    logging and reporting mechanisms may be chosen. The
                    mapping of event types to priorities is
                    vendor dependent. Vendors may also choose to allow
                    the user to control that mapping through proprietary
                    means. Table entries MUST persist across reboots for
                    CMTS devices and MUST NOT persist across reboots for CM
                    devices."
            ::= { docsDevEvent 7 }

    docsDevEvControlEntry OBJECT-TYPE
            SYNTAX   DocsDevEvControlEntry
            MAX-ACCESS   not-accessible
            STATUS   current
            DESCRIPTION
                "Allows configuration of the reporting mechanisms for a
                 particular event priority."
            INDEX { docsDevEvPriority }
            ::= { docsDevEvControlTable 1 }

    DocsDevEvControlEntry ::= SEQUENCE {
            docsDevEvPriority   INTEGER,
            docsDevEvReporting   BITS
            }

    docsDevEvPriority OBJECT-TYPE
            SYNTAX INTEGER {
                emergency(1),
                alert(2),
                critical(3),
                error(4),
                warning(5),
                notice(6),
                information(7),
                debug(8)
            }
            MAX-ACCESS   not-accessible
            STATUS   current
            DESCRIPTION
                "The priority level that is controlled by this
                 entry. These are ordered from most (emergency) to least
                 (debug) critical. Each event with a CM or CMTS has a
                 particular priority level associated with it (as defined
                 by the vendor).

                 emergency(1) events indicate vendor-specific fatal
                 hardware or software errors that prevent normal system
                 operation.

               alert(2) events indicate a serious failure that causes
               the reporting system to reboot but is not caused by
               hardware or software malfunctioning.

               critical(3) events indicate a serious failure that
               requires attention and prevents the device from
               transmitting data but that could be recovered without
               rebooting the system.

               error(4) and warning(5) events indicate that a failure
               occurred that could interrupt the normal data flow but
               that does not cause the device to re-register.

               notice(6) and information(7) events indicate a
               milestone or checkpoint in normal operation that could
               be of particular importance for troubleshooting.

               debug(8) events are reserved for vendor-specific
               events.

               During normal operation, no event more
               critical than notice(6) should be generated. Events
               between warning and emergency should be generated at
               appropriate levels of problems (e.g., emergency when the
               box is about to crash)."
          ::= { docsDevEvControlEntry 1 }

    docsDevEvReporting OBJECT-TYPE
          SYNTAX BITS {
              local(0),
              traps(1),
              syslog(2),
              -- The following are extensions to the original set of
              -- labels. The extensions start at an octet boundary.
              -- So for bits 3 - 7, one MUST set them to zero on send
              -- and one MUST ignore them on receipt.
              localVolatile(8),
              stdInterface(9)
          }
          MAX-ACCESS read-write
          STATUS   current
          DESCRIPTION
              "Defines the action to be taken on occurrence of this
               event class. Implementations may not necessarily
               support all options for all event classes but at
               minimum must allow traps and syslogging to be
               disabled.

                    If the local(0) bit is set, then log to the internal
                    log and update non-volatile store, for backward
                    compatibility with the original RFC 2669 definition.
                    If the traps(1) bit is set, then generate
                    an SNMP trap; if the syslog(2) bit is set, then
                    send a syslog message (assuming that the syslog address
                    is set). If the localVolatile(8) bit is set, then
                    log to the internal log without updating non-volatile
                    store. If the stdInterface(9) bit is set, then the
                    agent ignores all other bits except the local(0),
                    syslog(2), and localVolatile(8) bits. Setting the
                    stdInterface(9) bit indicates that RFC3413 and
                    RFC3014 are being used to control event reporting
                    mechanisms."
            ::= { docsDevEvControlEntry 2 }

    docsDevEventTable OBJECT-TYPE
            SYNTAX   SEQUENCE OF DocsDevEventEntry
            MAX-ACCESS not-accessible
            STATUS   current
            DESCRIPTION
                "Contains a log of network and device events that may be
                 of interest in fault isolation and troubleshooting.
                 If the local(0) bit is set in docsDevEvReporting,
                 entries in this table MUST persist across reboots."
            ::= { docsDevEvent 8 }

    docsDevEventEntry OBJECT-TYPE
            SYNTAX   DocsDevEventEntry
            MAX-ACCESS not-accessible
            STATUS   current
            DESCRIPTION
                "Describes a network or device event that may be of
                 interest in fault isolation and troubleshooting.
                 Multiple sequential identical events are represented by
                 incrementing docsDevEvCounts and setting
                 docsDevEvLastTime to the current time rather than
                 creating multiple rows.

                 Entries are created with the first occurrence of an
                 event. docsDevEvControl can be used to clear the
                 table. Individual events cannot be deleted."
            INDEX { docsDevEvIndex }
            ::= { docsDevEventTable 1 }

    DocsDevEventEntry ::= SEQUENCE {
            docsDevEvIndex   Integer32,
            docsDevEvFirstTime   DateAndTime,

```
            docsDevEvLastTime          DateAndTime,
            docsDevEvCounts            Counter32,
            docsDevEvLevel             INTEGER,
            docsDevEvId                Unsigned32,
            docsDevEvText              SnmpAdminString
        }

docsDevEvIndex OBJECT-TYPE
        SYNTAX      Integer32 (1..2147483647)
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "Provides relative ordering of the objects in the event
             log.  This object will always increase except when
             (a) the log is reset via docsDevEvControl,
             (b) the device reboots and does not implement
             non-volatile storage for this log, or (c) it reaches
             the value 2^31.  The next entry for all the above
             cases is 1."
        ::= { docsDevEventEntry 1 }

docsDevEvFirstTime OBJECT-TYPE
        SYNTAX      DateAndTime
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The value of docsDevDateTime at the time this entry was
             created."
        ::= { docsDevEventEntry 2 }

docsDevEvLastTime OBJECT-TYPE
        SYNTAX      DateAndTime
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "When an entry reports only one event, this object will
             have the same value as the corresponding instance of
             docsDevEvFirstTime.  When an entry reports multiple
             events, this object will record the value that
             docsDevDateTime had when the most recent event for this
             entry occurred."
        ::= { docsDevEventEntry 3 }

-- This object was renamed from docsDevEvCount to meet naming
-- requirements for Counter32
docsDevEvCounts OBJECT-TYPE
        SYNTAX      Counter32
        UNITS       "events"
```

```
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                "The number of consecutive event instances reported by
                 this entry.  This starts at 1 with the creation of this
                 row and increments by 1 for each subsequent duplicate
                 event."
            ::= { docsDevEventEntry 4 }
    docsDevEvLevel OBJECT-TYPE
            SYNTAX INTEGER {
                emergency(1),
                alert(2),
                critical(3),
                error(4),
                warning(5),
                notice(6),
                information(7),
                debug(8)
            }
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                "The priority level of this event, as defined by the
                 vendor.  These are ordered from most serious (emergency)
                 to least serious (debug).

                 emergency(1) events indicate vendor-specific fatal
                 hardware or software errors that prevent normal system
                 operation.

                 alert(2) events indicate a serious failure that causes
                 the reporting system to reboot but that is not caused by
                 hardware or software malfunctioning.

                 critical(3) events indicate a serious failure that
                 requires attention and prevents the device from
                 transmitting data but that could be recovered without
                 rebooting the system.

                 error(4) and warning(5) events indicate that a failure
                 occurred that could interrupt the normal data flow but
                 that does not cause the device to re-register.

                 notice(6) and information(7) events indicate a
                 milestone or checkpoint in normal operation that could
                 be of particular importance for troubleshooting.

                 debug(8) events are reserved for vendor-specific
```

            events.

            During normal operation, no event more
            critical than notice(6) should be generated. Events
            between warning and emergency should be generated at
            appropriate levels of problems (e.g., emergency when the
            box is about to crash)."
        ::= { docsDevEventEntry 5 }

   --
   -- It is strongly recommended that implementors follow the CableLabs
   -- enumerations for docsDevEvId, per the DOCSIS OSSIv1.1 spec
   -- and follow-on specifications.
   --

   docsDevEvId OBJECT-TYPE
        SYNTAX   Unsigned32
        MAX-ACCESS   read-only
        STATUS   current
        DESCRIPTION
            "For this product, uniquely identifies the type of event
             that is reported by this entry."
        REFERENCE
            "DOCSIS OSSI 1.1 Specification, Appendix H and
             DOCSIS OSSI 2.0 Specification, Annex D."
        ::= { docsDevEventEntry 6 }

   docsDevEvText OBJECT-TYPE
        SYNTAX   SnmpAdminString
        MAX-ACCESS   read-only
        STATUS   current
        DESCRIPTION
            "Provides a human-readable description of the event,
             including all relevant context (interface numbers,
             etc.)."
        ::= { docsDevEventEntry 7 }

   docsDevEvSyslogAddressType OBJECT-TYPE
        SYNTAX   InetAddressType
        MAX-ACCESS   read-write
        STATUS   current
        DESCRIPTION
            "The type of address of docsDevEvSyslogAddress. If
             no syslog server exists, this value should return
             unknown(0)."
        DEFVAL { unknown }
        ::= { docsDevEvent 9 }

       docsDevEvSyslogAddress OBJECT-TYPE
              SYNTAX   InetAddress
              MAX-ACCESS   read-write
              STATUS   current
              DESCRIPTION
                  "The Internet address of the Syslog server, as provided
                   by DHCP option 7 or set via SNMP management. If the
                   address of the server is set to the zero-length
                   string, the 0.0.0.0 IPv4 address, or the 0: IPv6
                   address, Syslog transmission is inhibited.

                   Note that if multiple values are provided to the CM in
                   DHCP option 7, the value of this MIB object MUST be the
                   first Syslog server address received.

                   By default at agent boot, this object returns the zero
                   length string."
              ::= { docsDevEvent 10 }

       docsDevEvThrottleThresholdExceeded OBJECT-TYPE
              SYNTAX TruthValue
              MAX-ACCESS read-only
              STATUS current
              DESCRIPTION
                  "If true(1), trap and syslog transmission is currently
                   inhibited due to exceeding the trap/syslog event
                   threshold in the current interval."
              ::= { docsDevEvent 11 }

       --
       -- Link Level Control Filtering
       --

       docsDevFilter OBJECT IDENTIFIER ::= { docsDevMIBObjects 6 }

       docsDevFilterLLCUnmatchedAction OBJECT-TYPE
              SYNTAX INTEGER {
                  discard(1),
                  accept(2)
              }
              MAX-ACCESS read-write
              STATUS   current
              DESCRIPTION
                  "LLC (Link Level Control) filters can be defined on an
                   inclusive or exclusive basis: CMs can be configured to
                   forward only packets matching a set of layer three
                   protocols, or to drop packets matching a set of layer
                   three protocols. Typical use of these filters is to

                    filter out possibly harmful (given the context of a
                    large metropolitan LAN) protocols.

                    If set to discard(1), any L2 packet that does not match
                    at least one filter in the docsDevFilterLLCTable will be
                    discarded. If set to accept(2), any L2 packet that
                    does not match at least one filter in the
                    docsDevFilterLLCTable will be accepted for further
                    processing (e.g., bridging). In other words, if the
                    packet does not match an entry in the table, it takes
                    this action; if it does match an entry in the table, it
                    takes the opposite of this action."
            DEFVAL { accept }
            ::= { docsDevFilter 1 }

    docsDevFilterLLCTable OBJECT-TYPE
            SYNTAX   SEQUENCE OF DocsDevFilterLLCEntry
            MAX-ACCESS   not-accessible
            STATUS   current
            DESCRIPTION
                "A list of filters to apply to (bridged) LLC
                 traffic. The filters in this table are applied to
                 incoming traffic on the appropriate interface(s) prior
                 to any further processing (e.g., before the packet
                 is handed off for level 3 processing, or for bridging).
                 The specific action taken when no filter is matched is
                 controlled by docsDevFilterLLCUnmatchedAction. Table
                 entries MUST NOT persist across reboots for any device."
            ::= { docsDevFilter 2 }

    docsDevFilterLLCEntry OBJECT-TYPE
            SYNTAX   DocsDevFilterLLCEntry
            MAX-ACCESS   not-accessible
            STATUS   current
            DESCRIPTION
                "Describes a single filter to apply to (bridged) LLC
                 traffic received on a specified interface. "
            INDEX { docsDevFilterLLCIndex }
            ::= { docsDevFilterLLCTable 1 }

    DocsDevFilterLLCEntry ::= SEQUENCE {
            docsDevFilterLLCIndex   Integer32,
            docsDevFilterLLCStatus   RowStatus,
            docsDevFilterLLCIfIndex   InterfaceIndexOrZero,
            docsDevFilterLLCProtocolType   INTEGER,
            docsDevFilterLLCProtocol   Integer32,
            docsDevFilterLLCMatches   Counter32
            }

docsDevFilterLLCIndex OBJECT-TYPE
        SYNTAX   Integer32 (1..2147483647)
        MAX-ACCESS   not-accessible
        STATUS   current
        DESCRIPTION
            "Index used for the identification of filters (note that
            LLC filter order is irrelevant)."
        ::= { docsDevFilterLLCEntry 1 }
docsDevFilterLLCStatus OBJECT-TYPE
        SYNTAX   RowStatus
        MAX-ACCESS   read-create
        STATUS   current
        DESCRIPTION
            "Controls and reflects the status of rows in this
            table. There is no restriction on changing any of the
            associated columns for this row while this object is set
            to active.

            Specifying only this object (with the
            appropriate index) on a CM is sufficient to create a
            filter row that matches all inbound packets on the
            ethernet interface and results in the packets being
            discarded. docsDevFilterLLCIfIndex (at least) must be
            specified on a CMTS to create a row."
        ::= { docsDevFilterLLCEntry 2}

docsDevFilterLLCIfIndex OBJECT-TYPE
        SYNTAX   InterfaceIndexOrZero
        MAX-ACCESS   read-create
        STATUS   current
        DESCRIPTION
            "The entry interface to which this filter applies. The
            value corresponds to ifIndex for either a CATV MAC or
            another network interface. If the value is zero, the
            filter applies to all interfaces. In Cable Modems, the
            default value is the customer side interface(s). In
            CMTSs, this object has to be specified to
            create a row in this table.

            Note that according to the DOCSIS OSSIv1.1
            specification, ifIndex '1' in the CM means that this
            row applies to all Cable Modem-to-CPE Interfaces
            (CMCI)."
        REFERENCE
            "DOCSIS OSSI 1.1 Specification, Section 3.3.4.1. and
            DOCSIS OSSI 2.0 Specification, Section 6.3.4.1."
        ::= { docsDevFilterLLCEntry 3 }

```
    docsDevFilterLLCProtocolType OBJECT-TYPE
            SYNTAX INTEGER {
                ethertype(1),
                dsap(2)
            }
            MAX-ACCESS  read-create
            STATUS      current
            DESCRIPTION
                "The format of the value in docsDevFilterLLCProtocol:
                 either a two-byte Ethernet Ethertype, or a one-byte
                 802.2 Service Access Point (SAP) value.  ethertype(1)
                 also applies to Standard Network Access Protocol
                 (SNAP) encapsulated frames."
            DEFVAL { ethertype }
            ::= { docsDevFilterLLCEntry 4 }

    docsDevFilterLLCProtocol OBJECT-TYPE
            SYNTAX      Integer32 (0..65535)
            MAX-ACCESS  read-create
            STATUS      current
            DESCRIPTION
                "The layer-three protocol for which this filter applies.
                 The protocol value format depends on
                 docsDevFilterLLCProtocolType.  Note that for SNAP
                 frames, ethertype filtering is performed rather than
                 Destination Service Access Point (DSAP) =0xAA."
            DEFVAL { 0 }
            ::= { docsDevFilterLLCEntry 5 }

    docsDevFilterLLCMatches OBJECT-TYPE
            SYNTAX      Counter32
            UNITS       "matches"
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                "Counts the number of times this filter was matched."
            ::= { docsDevFilterLLCEntry 6 }

    --
    -- IPv4 Filtering
    --

    docsDevFilterIpDefault OBJECT-TYPE
            SYNTAX INTEGER {
                discard(1),
                accept(2)
            }
            MAX-ACCESS  read-write
```

```
        STATUS      deprecated
        DESCRIPTION
            "The default behavior for (bridged) packets that do not
             match IP filters (or Internet filters, if implemented)
             is defined by docsDevFilterIpDefault.

             If set to discard(1), all packets not matching an IP
             filter in docsDevFilterIpTable will be discarded.  If
             set to accept(2), all packets not matching an IP filter
             or an Internet filter will be accepted for further
             processing (e.g., bridging)."
        DEFVAL { accept }
        ::= { docsDevFilter 3 }

 docsDevFilterIpTable OBJECT-TYPE
        SYNTAX      SEQUENCE OF DocsDevFilterIpEntry
        MAX-ACCESS  not-accessible
        STATUS      deprecated
        DESCRIPTION
            "An ordered list of filters or classifiers to apply to
             IP traffic.  Filter application is ordered by the filter
             index, rather than by a best match algorithm (note that
             this implies that the filter table may have gaps in the
             index values).  Packets that match no filters will have
             policy 0 in the docsDevFilterPolicyTable applied to
             them, if it exists.  Otherwise, Packets that match no
             filters are discarded or forwarded according to the
             setting of docsDevFilterIpDefault.

             Any IP packet can theoretically match multiple rows of
             this table.  When considering a packet, the table is
             scanned in row index order (e.g., filter 10 is checked
             before filter 20).  If the packet matches that filter
             (which means that it matches ALL criteria for that row),
             actions appropriate to docsDevFilterIpControl and
             docsDevFilterPolicyId are taken.  If the packet was
             discarded processing is complete.  If
             docsDevFilterIpContinue is set to true, the filter
             comparison continues with the next row in the table,
             looking for additional matches.

             If the packet matches no filter in the table, the packet
             is accepted or dropped for further processing
             according to the setting of docsDevFilterIpDefault.
             If the packet is accepted, the actions specified by
             policy group 0 (e.g., the rows in
             docsDevFilterPolicyTable that have a value of 0 for
             docsDevFilterPolicyId) are taken, if that policy
```

                group exists.

                Logically, this table is consulted twice during the
                processing of any IP packet: once upon its acceptance
                from the L2 entity, and once upon its transmission to
                the L2 entity. In actuality, for cable modems, IP
                filtering is generally the only IP processing done for
                transit traffic. This means that inbound and outbound
                filtering can generally be done at the same time with
                one pass through the filter table.

                The objects in this table are only accessible from cable
                devices that are not operating in DiffServ MIB mode
                (RFC 3289). See the conformance section for details.

                Note that some devices are required by other
                specifications (e.g., the DOCSIS OSSIv1.1 specification)
                to support the legacy SNMPv1/v2c docsDevFilter mode
                for backward compatibility.

                Table entries MUST NOT persist across reboots for any
                device.

                This table is deprecated. Instead, use the DiffServ MIB
                from RFC 3289."
            ::= { docsDevFilter 4 }

    docsDevFilterIpEntry OBJECT-TYPE
            SYNTAX   DocsDevFilterIpEntry
            MAX-ACCESS   not-accessible
            STATUS   deprecated
            DESCRIPTION
                "Describes a filter to apply to IP traffic received on a
                specified interface. All identity objects in this table
                (e.g., source and destination address/mask, protocol,
                source/dest port, TOS/mask, interface and direction)
                must match their respective fields in the packet for
                any given filter to match.

                To create an entry in this table, docsDevFilterIpIfIndex
                must be specified."
            INDEX { docsDevFilterIpIndex }
            ::= { docsDevFilterIpTable 1 }

    DocsDevFilterIpEntry ::= SEQUENCE {
            docsDevFilterIpIndex   Integer32,
            docsDevFilterIpStatus   RowStatus,
            docsDevFilterIpControl   INTEGER,

```
            docsDevFilterIpIfIndex              InterfaceIndexOrZero,
            docsDevFilterIpDirection            INTEGER,
            docsDevFilterIpBroadcast            TruthValue,
            docsDevFilterIpSaddr                IpAddress,
            docsDevFilterIpSmask                IpAddress,
            docsDevFilterIpDaddr                IpAddress,
            docsDevFilterIpDmask                IpAddress,
            docsDevFilterIpProtocol             Integer32,
            docsDevFilterIpSourcePortLow        Integer32,
            docsDevFilterIpSourcePortHigh       Integer32,
            docsDevFilterIpDestPortLow          Integer32,
            docsDevFilterIpDestPortHigh         Integer32,
            docsDevFilterIpMatches              ZeroBasedCounter32,
            docsDevFilterIpTos                  OCTET STRING,
            docsDevFilterIpTosMask              OCTET STRING,
            docsDevFilterIpContinue             TruthValue,
            docsDevFilterIpPolicyId             Integer32
        }

docsDevFilterIpIndex OBJECT-TYPE
        SYNTAX       Integer32 (1..2147483647)
        MAX-ACCESS   not-accessible
        STATUS       deprecated
        DESCRIPTION
            "Index used to order the application of filters.
             The filter with the lowest index is always applied
             first."
        ::= { docsDevFilterIpEntry 1 }

docsDevFilterIpStatus OBJECT-TYPE
        SYNTAX       RowStatus
        MAX-ACCESS   read-create
        STATUS       deprecated
        DESCRIPTION
            "Controls and reflects the status of rows in this
             table.  Specifying only this object (with the
             appropriate index) on a CM is sufficient to create a
             filter row that matches all inbound packets on the
             ethernet interface and results in the packets being
             discarded.  docsDevFilterIpIfIndex (at least) must be
             specified on a CMTS to create a row.  Creation of the
             rows may be done via either create-and-wait or
             create-and-go, but the filter is not applied until this
             object is set to (or changes to) active.  There is no
             restriction in changing any object in a row while this
             object is set to active."
        ::= { docsDevFilterIpEntry 2 }
```

```
docsDevFilterIpControl OBJECT-TYPE
        SYNTAX INTEGER {
            discard(1),
            accept(2),
            policy(3)
        }
        MAX-ACCESS  read-create
        STATUS      deprecated
        DESCRIPTION
            "If set to discard(1), all packets matching this filter
             will be discarded, and scanning of the remainder of the
             filter list will be aborted.  If set to accept(2), all
             packets matching this filter will be accepted for
             further processing (e.g., bridging).  If
             docsDevFilterIpContinue is set to true, see if there
             are other matches; otherwise, done.  If set to
             policy (3), execute the policy entries
             matched by docsDevFilterIpPolicyId in
             docsDevFilterPolicyTable.

             If docsDevFilterIpContinue is set to true, continue
             scanning the table for other matches; otherwise, done."
        DEFVAL { discard }
        ::= { docsDevFilterIpEntry 3 }

docsDevFilterIpIfIndex OBJECT-TYPE
        SYNTAX      InterfaceIndexOrZero
        MAX-ACCESS  read-create
        STATUS      deprecated
        DESCRIPTION
            "The entry interface to which this filter applies.  The
             value corresponds to ifIndex for either a CATV MAC or
             another interface.  If the value is zero, the
             filter applies to all interfaces.  Default value in CMs
             is the index of the customer-side (e.g., ethernet)
             interface(s).  In CMTSes, this object MUST be
             specified to create a row in this table.

             Note that according to the DOCSIS OSSIv1.1
             specification, ifIndex '1' in the Cable Modem means
             that this row applies to all CMCI (customer-facing)
             interfaces."
        REFERENCE
            "DOCSIS OSSI 1.1 Specification, Section 3.3.4.1. and
             DOCSIS OSSI 2.0 Specification, Section 6.3.4.1."
        ::= { docsDevFilterIpEntry 4 }

docsDevFilterIpDirection OBJECT-TYPE
```

```
        SYNTAX INTEGER {
            inbound(1),
            outbound(2),
            both(3)
        }
        MAX-ACCESS  read-create
        STATUS      deprecated
        DESCRIPTION
            "Determines whether the filter is applied to inbound(1)
             traffic, outbound(2) traffic, or traffic in both(3)
             directions."
        DEFVAL { inbound }
        ::= { docsDevFilterIpEntry 5 }

    docsDevFilterIpBroadcast OBJECT-TYPE
        SYNTAX      TruthValue
        MAX-ACCESS  read-create
        STATUS      deprecated
        DESCRIPTION
            "If set to true(1), the filter only applies to multicast
             and broadcast traffic.  If set to false(2), the filter
             applies to all traffic."
        DEFVAL { false }
        ::= { docsDevFilterIpEntry 6 }

    docsDevFilterIpSaddr OBJECT-TYPE
        SYNTAX      IpAddress
        MAX-ACCESS  read-create
        STATUS      deprecated
        DESCRIPTION
            "The source IP address, or portion thereof, that is to be
             matched for this filter.  The source address is first
             masked (ANDed) against docsDevFilterIpSmask before
             being compared to this value.  A value of 0 for this
             object and 0 for the mask matches all IP addresses."
        DEFVAL { '00000000'h }
        ::= { docsDevFilterIpEntry 7 }

    docsDevFilterIpSmask OBJECT-TYPE
        SYNTAX      IpAddress
        MAX-ACCESS  read-create
        STATUS      deprecated
        DESCRIPTION
            "A bit mask that is to be applied to the source address
             prior to matching.  This mask is not necessarily the
             same as a subnet mask, but 1s bits must be leftmost and
             contiguous."
        DEFVAL { '00000000'h }
```

                   ::= { docsDevFilterIpEntry 8 }

    docsDevFilterIpDaddr OBJECT-TYPE
            SYNTAX   IpAddress
            MAX-ACCESS   read-create
            STATUS   deprecated
            DESCRIPTION
                "The destination IP address, or portion thereof, that is
                 to be matched for this filter. The destination address
                 is first masked (ANDed) against docsDevFilterIpDmask
                 before being compared to this value. A value of
                 00000000 for this object and 00000000 for the mask
                 matches all IP addresses."
            DEFVAL { '00000000'h }
            ::= { docsDevFilterIpEntry 9 }

    docsDevFilterIpDmask OBJECT-TYPE
            SYNTAX   IpAddress
            MAX-ACCESS   read-create
            STATUS   deprecated
            DESCRIPTION
                "A bit mask that is to be applied to the destination
                 address prior to matching. This mask is not necessarily
                 the same as a subnet mask, but 1s bits MUST be leftmost
                 and contiguous."
            DEFVAL { '00000000'h }
            ::= { docsDevFilterIpEntry 10 }

    docsDevFilterIpProtocol OBJECT-TYPE
            SYNTAX Integer32 (0..256)
            MAX-ACCESS   read-create
            STATUS   deprecated
            DESCRIPTION
                "The IP protocol value that is to be matched. For
                 example, icmp is 1, tcp is 6, and udp is 17. A value of
                 256 matches ANY protocol."
            REFERENCE "www.iana.org/assignments/protocol-numbers"
            DEFVAL { 256 }
            ::= { docsDevFilterIpEntry 11 }

    docsDevFilterIpSourcePortLow OBJECT-TYPE
            SYNTAX   Integer32 (0..65535)
            MAX-ACCESS   read-create
            STATUS   deprecated
            DESCRIPTION
                "This is the inclusive lower bound of the transport-layer
                 source port range that is to be matched. If the IP
                 protocol of the packet is neither UDP nor TCP, this

```
                object is ignored during matching."
         REFERENCE "www.iana.org/assignments/port-numbers"
         DEFVAL { 0 }
         ::= { docsDevFilterIpEntry 12 }

   docsDevFilterIpSourcePortHigh OBJECT-TYPE
         SYNTAX       Integer32 (0..65535)
         MAX-ACCESS  read-create
         STATUS       deprecated
         DESCRIPTION
             "This is the inclusive upper bound of the transport-layer
              source port range that is to be matched.  If the IP
              protocol of the packet is neither UDP nor TCP, this
              object is ignored during matching."
         REFERENCE "www.iana.org/assignments/port-numbers"
         DEFVAL { 65535 }
         ::= { docsDevFilterIpEntry 13 }

   docsDevFilterIpDestPortLow OBJECT-TYPE
         SYNTAX       Integer32 (0..65535)
         MAX-ACCESS  read-create
         STATUS       deprecated
         DESCRIPTION
             "This is the inclusive lower bound of the transport-layer
              destination port range that is to be matched.  If the IP
              protocol of the packet is neither UDP nor TCP, this
              object is ignored during matching."
         REFERENCE "www.iana.org/assignments/port-numbers"
         DEFVAL { 0 }
         ::= { docsDevFilterIpEntry 14 }

   docsDevFilterIpDestPortHigh OBJECT-TYPE
         SYNTAX       Integer32 (0..65535)
         MAX-ACCESS  read-create
         STATUS       deprecated
         DESCRIPTION
             "This is the inclusive upper bound of the transport-layer
              destination port range that is to be matched.  If the IP
              protocol of the packet is neither UDP nor TCP, this
              object is ignored during matching."
         REFERENCE "www.iana.org/assignments/port-numbers"
         DEFVAL { 65535 }
         ::= { docsDevFilterIpEntry 15 }

   docsDevFilterIpMatches OBJECT-TYPE
         SYNTAX       ZeroBasedCounter32
         UNITS        "matches"
         MAX-ACCESS  read-only
```

```
            STATUS       deprecated
            DESCRIPTION
                "Counts the number of times this filter was matched.
                 This object is initialized to 0 at boot, or at row
                 creation, and is reset only upon reboot."
            ::= { docsDevFilterIpEntry 16 }

    docsDevFilterIpTos  OBJECT-TYPE
            SYNTAX       OCTET STRING (SIZE (1))
            MAX-ACCESS   read-create
            STATUS       deprecated
            DESCRIPTION
                "This is the value to be matched to the packet's
                 TOS (Type of Service) value (after the TOS value
                 is ANDed with docsDevFilterIpTosMask).  A value for this
                 object of 0 and a mask of 0 matches all TOS values."
            DEFVAL { '00'h }
            ::= { docsDevFilterIpEntry 17 }

    docsDevFilterIpTosMask OBJECT-TYPE
            SYNTAX       OCTET STRING (SIZE (1))
            MAX-ACCESS   read-create
            STATUS       deprecated
            DESCRIPTION
                "The mask to be applied to the packet's TOS value before
                 matching."
            DEFVAL { '00'h }
            ::= { docsDevFilterIpEntry 18 }

    docsDevFilterIpContinue OBJECT-TYPE
            SYNTAX       TruthValue
            MAX-ACCESS   read-create
            STATUS       deprecated
            DESCRIPTION
                "If this value is set to true and docsDevFilterIpControl
                 is anything but discard (1), continue scanning and
                 applying policies.  See Section 3.3.3 for more
                 details."
            DEFVAL { false }
            ::= { docsDevFilterIpEntry 19 }

    docsDevFilterIpPolicyId OBJECT-TYPE
            SYNTAX       Integer32 (0..2147483647)
            MAX-ACCESS   read-create
            STATUS       deprecated
            DESCRIPTION
                "This object points to an entry in
                 docsDevFilterPolicyTable.  If docsDevFilterIpControl
```

                    is set to policy (3), execute all matching policies
                    in docsDevFilterPolicyTable. If no matching policy
                    exists, treat as if docsDevFilterIpControl were set
                    to accept (1). If this object is set to the value of
                    0, there is no matching policy, and
                    docsDevFilterPolicyTable MUST NOT be consulted."
               DEFVAL { 0 }
               ::= { docsDevFilterIpEntry 20 }

        --
        -- Policy Mapping Table
        --

        docsDevFilterPolicyTable OBJECT-TYPE
               SYNTAX   SEQUENCE OF DocsDevFilterPolicyEntry
               MAX-ACCESS   not-accessible
               STATUS   deprecated
               DESCRIPTION
                    "A Table that maps between a policy group ID and a set
                    of pointers to policies to be applied. All rows with
                    the same docsDevFilterPolicyId are part of the same
                    group of policy pointers and are applied in the order
                    in this table. docsDevFilterPolicyTable exists to
                    allow multiple policy actions (referenced by policy
                    pointers) to be applied to any given classified packet.
                    The policy actions are applied in index order.
                    For example:

                    Index   ID   Type   Action
                     1   1   TOS   1
                     9   5   TOS   1
                     12   1   IPSEC   3

                    This says that a packet that matches a filter with
                    policy id 1 first has TOS policy 1 applied (which might
                    set the TOS bits to enable a higher priority) and next
                    has the IPSEC policy 3 applied (which may result in the
                    packets being dumped into a secure VPN to a remote
                    encryptor).

                    Policy ID 0 is reserved for default actions and is
                    applied only to packets that match no filters in
                    docsDevFilterIpTable.

                    Table entries MUST NOT persist across reboots for any
                    device.

                    This table is deprecated. Instead, use the DiffServ MIB

```
            from RFC 3289."
        ::= { docsDevFilter 5 }

docsDevFilterPolicyEntry OBJECT-TYPE
        SYNTAX       DocsDevFilterPolicyEntry
        MAX-ACCESS   not-accessible
        STATUS       deprecated
        DESCRIPTION
            "An entry in the docsDevFilterPolicyTable.  Entries are
             created by Network Management.  To create an entry,
             docsDevFilterPolicyId MUST be specified."
        INDEX { docsDevFilterPolicyIndex }
        ::= { docsDevFilterPolicyTable 1 }

DocsDevFilterPolicyEntry ::= SEQUENCE {
        docsDevFilterPolicyIndex    Integer32,
        docsDevFilterPolicyId       Integer32,
--      docsDevFilterPolicyType     INTEGER,
--      docsDevFilterPolicyAction   Integer32,
        docsDevFilterPolicyStatus   RowStatus,
        docsDevFilterPolicyPtr      RowPointer
        }

docsDevFilterPolicyIndex OBJECT-TYPE
        SYNTAX       Integer32 (1..2147483647)
        MAX-ACCESS   not-accessible
        STATUS       deprecated
        DESCRIPTION "Index value for the table."
        ::= { docsDevFilterPolicyEntry 1 }

docsDevFilterPolicyId OBJECT-TYPE
        SYNTAX       Integer32 (0..2147483647)
        MAX-ACCESS   read-create
        STATUS       deprecated
        DESCRIPTION
            "Policy ID for this entry.  If a policy ID can apply to
             multiple rows of this table, all relevant policies are
             executed.  Policy 0 (if populated) is applied to all
             packets that do not match any of the filters.  N.B. If
             docsDevFilterIpPolicyId is set to 0, it DOES NOT match
             policy 0 of this table."
        ::= { docsDevFilterPolicyEntry 2 }

-- The following two objects were removed and never used; however,
-- to preserve OID numbering, they are simply commented out to
-- to ensure that they are not used again.
-- docsDevFilterPolicyType ::= { docsDevFilterPolicyEntry 3 }
-- docsDevFilterPolicyAction ::= { docsDevFilterPolicyEntry 4 }
```

docsDevFilterPolicyStatus OBJECT-TYPE
        SYNTAX   RowStatus
        MAX-ACCESS   read-create
        STATUS   deprecated
        DESCRIPTION
            "Object used to create an entry in this table. There is
             no restriction in changing any object in a row while
             this object is set to active.
             The following object MUST have a valid value before this
             object can be set to active: docsDevFilterPolicyPtr."
        ::= { docsDevFilterPolicyEntry 5 }

docsDevFilterPolicyPtr OBJECT-TYPE
        SYNTAX   RowPointer
        MAX-ACCESS   read-create
        STATUS   deprecated
        DESCRIPTION
            "This object points to a row in an applicable filter
             policy table. Currently, the only standard policy
             table is docsDevFilterTosTable.

             Per the textual convention, this object points to the
             first accessible object in the row; e.g., to point to a
             row in docsDevFilterTosTable with an index of 21, the
             value of this object would be the object identifier
             docsDevTosStatus.21.

             Vendors are recommended to adhere to the same convention
             when adding vendor-specific policy table extensions.

             If this pointer references an empty or non-existent
             row, then no policy action is taken.

             The default upon row creation is a null pointer that
             results in no policy action being taken."
        DEFVAL { zeroDotZero }
        ::= { docsDevFilterPolicyEntry 6 }

    --
    -- TOS Policy action table
    --

    docsDevFilterTosTable OBJECT-TYPE
        SYNTAX   SEQUENCE OF DocsDevFilterTosEntry
        MAX-ACCESS   not-accessible
        STATUS   deprecated
        DESCRIPTION
            "Table used to describe Type of Service (TOS) bits

              processing.

              This table is an adjunct to the docsDevFilterIpTable
              and the docsDevFilterPolicy table. Entries in the
              latter table can point to specific rows in this (and
              other) tables and cause specific actions to be taken.
              This table permits the manipulation of the value of the
              Type of Service bits in the IP header of the matched
              packet as follows:

              Set the tosBits of the packet to
                 (tosBits & docsDevFilterTosAndMask) |
                 docsDevFilterTosOrMask

              This construct allows you to do a clear and set of all
              the TOS bits in a flexible manner.

              Table entries MUST NOT persist across reboots for any
              device.

              This table is deprecated. Instead, use the DiffServ MIB
              from RFC 3289."
           ::= { docsDevFilter 6 }

   docsDevFilterTosEntry OBJECT-TYPE
           SYNTAX   DocsDevFilterTosEntry
           MAX-ACCESS   not-accessible
           STATUS   deprecated
           DESCRIPTION
              "A TOS policy entry."
           INDEX { docsDevFilterTosIndex }
           ::= { docsDevFilterTosTable 1 }

   DocsDevFilterTosEntry ::= SEQUENCE {
              docsDevFilterTosIndex   Integer32,
              docsDevFilterTosStatus RowStatus,
              docsDevFilterTosAndMask OCTET STRING,
              docsDevFilterTosOrMask OCTET STRING
           }

   docsDevFilterTosIndex OBJECT-TYPE
           SYNTAX   Integer32 (1..2147483647)
           MAX-ACCESS   not-accessible
           STATUS   deprecated
           DESCRIPTION
              "The unique index for this row. There are no ordering
               requirements for this table, and any valid index may be
               specified."

```
             ::= { docsDevFilterTosEntry 1 }

    docsDevFilterTosStatus OBJECT-TYPE
            SYNTAX        RowStatus
            MAX-ACCESS    read-create
            STATUS        deprecated
            DESCRIPTION
                "The object used to create and delete entries in this
                 table.  A row created by specifying just this object
                 results in a row that specifies no change to the TOS
                 bits.  A row may be created using either the
                 create-and-go or create-and-wait paradigms.  There is
                 no restriction on the ability to change values in this
                 row while the row is active."
            ::= { docsDevFilterTosEntry 2 }

    docsDevFilterTosAndMask OBJECT-TYPE
            SYNTAX        OCTET STRING (SIZE (1))
            MAX-ACCESS    read-create
            STATUS        deprecated
            DESCRIPTION
                "This value is bitwise ANDed with the matched packet's
                 TOS bits."
            DEFVAL { 'ff'h }
            ::= { docsDevFilterTosEntry 3 }

    docsDevFilterTosOrMask OBJECT-TYPE
            SYNTAX        OCTET STRING (SIZE (1))
            MAX-ACCESS    read-create
            STATUS        deprecated
            DESCRIPTION
                "This value is bitwise ORed with the result from the
                 AND procedure (tosBits & docsDevFilterTosAndMask).
                 The result then replaces the packet's TOS bits."
            DEFVAL { '00'h }
            ::= { docsDevFilterTosEntry 4 }

    --
    -- CPE IP Management and anti-spoofing group.  Only implemented on
    -- Cable Modems.
    --

    docsDevCpe OBJECT IDENTIFIER ::= { docsDevMIBObjects 7 }

    docsDevCpeEnroll OBJECT-TYPE
            SYNTAX        INTEGER {
                none(1),
                any(2)
```

```
            }
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                "This object controls the population of
                 docsDevFilterCpeTable.
                 If set to none, the filters must be set manually
                 by a network management action (either configuration
                 or SNMP set).
                 If set to any, the CM wiretaps the packets originating
                 from the ethernet and enrolls up to docsDevCpeIpMax
                 addresses as based on the source IPv4 or v6 addresses of
                 those packets."
            DEFVAL { any }
            ::= { docsDevCpe 1 }

    docsDevCpeIpMax OBJECT-TYPE
            SYNTAX      Integer32 (-1..2147483647)
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                "This object controls the maximum number of CPEs allowed
                 to be learned behind this device.  If set to zero, any
                 number of CPEs may connect up to the maximum permitted
                 for the device.
                 If set to -1, no filtering is done on CPE source
                 addresses, and no entries are made in the
                 docsDevFilterCpeTable via learning.  If an attempt is
                 made to set this to a number greater than that
                 permitted for the device, it is set to that maximum."
            DEFVAL { -1 }
            ::= { docsDevCpe 2 }

    docsDevCpeTable OBJECT-TYPE
            SYNTAX      SEQUENCE OF DocsDevCpeEntry
            MAX-ACCESS  not-accessible
            STATUS      deprecated
            DESCRIPTION
                "This table lists the IPv4 addresses seen (or permitted)
                 as source addresses in packets originating from the
                 customer interface on this device.  In addition, this
                 table can be provisioned with the specific addresses
                 permitted for the CPEs via the normal row creation
                 mechanisms.  Table entries MUST NOT persist across
                 reboots for any device.

                 N.B.  Management action can add entries in this table
                 and in docsDevCpeIpTable past the value of
```

                    docsDevCpeIpMax. docsDevCpeIpMax ONLY restricts the
                    ability of the CM to add learned addresses
                    automatically.

                    This table is deprecated and is replaced by
                    docsDevCpeInetTable."
               ::= { docsDevCpe 3 }

       docsDevCpeEntry OBJECT-TYPE
               SYNTAX   DocsDevCpeEntry
               MAX-ACCESS not-accessible
               STATUS   deprecated
               DESCRIPTION
                    "An entry in the docsDevFilterCpeTable. There is one
                    entry for each IPv4 CPE seen or provisioned. If
                    docsDevCpeIpMax is set to -1, this table is ignored;
                    otherwise, upon receipt of an IP packet from the
                    customer interface of the CM, the source IP address is
                    checked against this table. If the address is in the
                    table, packet processing continues. If the address is
                    not in the table but docsDevCpeEnroll is set to any
                    and the sum of the table sizes of docsDevCpeTable and
                    docsDevCpeInetTable is less than docsDevCpeIpMax, the
                    address is added to the table, and packet processing
                    continues. Otherwise, the packet is dropped.

                    The filtering actions specified by this table occur
                    after any LLC filtering (docsDevFilterLLCTable), but
                    prior to any IP filtering (docsDevFilterIpTable,
                    docsDevNmAccessTable)."
               INDEX   { docsDevCpeIp }
               ::= {docsDevCpeTable 1 }

       DocsDevCpeEntry ::= SEQUENCE {
               docsDevCpeIp   IpAddress,
               docsDevCpeSource INTEGER,
               docsDevCpeStatus RowStatus
               }

       docsDevCpeIp OBJECT-TYPE
               SYNTAX   IpAddress
               MAX-ACCESS not-accessible
               STATUS   deprecated
               DESCRIPTION
                    "The IPv4 address to which this entry applies.

                    N.B. Attempts to set all zeros or all ones address
                    values MUST be rejected."

              ::= { docsDevCpeEntry 1 }

    docsDevCpeSource OBJECT-TYPE
            SYNTAX   INTEGER {
                other(1),
                manual(2),
                learned(3)
            }

            MAX-ACCESS read-only
            STATUS   deprecated
            DESCRIPTION
                "This object describes how this entry was created. If
                 the value is manual(2), this row was created by a
                 network management action (either configuration or
                 SNMP set). If set to learned(3), then it was found via
                 looking at the source IPv4 address of a received packet.
                 The value other(1) is used for any entries that do not
                 meet manual(2) or learned(3) criteria."
            ::= { docsDevCpeEntry 2 }

    docsDevCpeStatus OBJECT-TYPE
            SYNTAX RowStatus
            MAX-ACCESS read-create
            STATUS deprecated
            DESCRIPTION
                "Standard object to manipulate rows. To create a row in
                 this table, one only needs to specify this object.
                 Management stations SHOULD use the create-and-go
                 mechanism for creating rows in this table."
            ::= { docsDevCpeEntry 3 }

    --
    -- Internet CPE Management and anti spoofing group, for support of
    -- non-IPv4 CPEs.
    --

    docsDevCpeInetTable OBJECT-TYPE
            SYNTAX   SEQUENCE OF DocsDevCpeInetEntry
            MAX-ACCESS not-accessible
            STATUS   current
            DESCRIPTION
                "This table lists the IP addresses seen (or permitted) as
                 source addresses in packets originating from the
                 customer interface on this device. In addition, this
                 table can be provisioned with the specific addresses
                 permitted for the CPEs via the normal row creation
                 mechanisms.

            N.B. Management action can add entries in this table
            and in docsDevCpeIpTable past the value of
            docsDevCpeIpMax. docsDevCpeIpMax ONLY restricts the
            ability of the CM to add learned addresses
            automatically.

            Table entries MUST NOT persist across reboots for any
            device.

            This table exactly mirrors docsDevCpeTable and applies
            to IPv4 and IPv6 addresses."
        ::= { docsDevCpe 4 }

    docsDevCpeInetEntry OBJECT-TYPE
            SYNTAX   DocsDevCpeInetEntry
            MAX-ACCESS not-accessible
            STATUS   current
            DESCRIPTION
            "An entry in the docsDevFilterCpeInetTable. There is one
            entry for each IP CPE seen or provisioned. If
            docsDevCpeIpMax is set to -1, this table is ignored;
            otherwise, upon receipt of an IP packet from the
            customer interface of the CM, the source IP address is
            checked against this table. If the address is in the
            table, packet processing continues. If the address is
            not in the table but docsDevCpeEnroll is set to any and
            the sum of the table sizes for docsDevCpeTable and
            docsDevCpeInetTable is less than docsDevCpeIpMax, the
            address is added to the table, and packet processing
            continues. Otherwise, the packet is dropped.

            The filtering actions specified by this table occur
            after any LLC filtering (docsDevFilterLLCTable), but
            prior to any IP filtering (docsDevFilterIpTable,
            docsDevNmAccessTable).

            When an agent (cable modem) restarts, then all
            dynamically created rows are lost."
            INDEX   { docsDevCpeInetType, docsDevCpeInetAddr }
            ::= { docsDevCpeInetTable 1 }

    DocsDevCpeInetEntry ::= SEQUENCE {
            docsDevCpeInetType   InetAddressType,
            docsDevCpeInetAddr   InetAddress,
            docsDevCpeInetSource   INTEGER,
            docsDevCpeInetRowStatus RowStatus
            }

    docsDevCpeInetType OBJECT-TYPE
            SYNTAX   InetAddressType
            MAX-ACCESS not-accessible
            STATUS   current
            DESCRIPTION
                "The type of internet address of docsDevCpeInetAddr."
             ::= { docsDevCpeInetEntry 1 }

    docsDevCpeInetAddr OBJECT-TYPE
            SYNTAX   InetAddress
            MAX-ACCESS not-accessible
            STATUS   current
            DESCRIPTION
                "The Internet address to which this entry applies.

                 Implementors need to be aware that if the size of
                 docsDevCpeInetAddr exceeds 114 octets OIDs of
                 instances of columns in this row will have more
                 than 128 sub-identifiers and cannot be accessed
                 using SNMPv1, SNMPv2c, or SNMPv3. Only unicast
                 address are allowed for this object."
             ::= { docsDevCpeInetEntry 2 }

    docsDevCpeInetSource OBJECT-TYPE
            SYNTAX   INTEGER {
                manual(2),
                learned(3)
            }
            MAX-ACCESS read-only
            STATUS   current
            DESCRIPTION
                "This object describes how this entry was created. If
                 the value is manual(2), this row was created by a
                 network management action (either configuration or
                 SNMP set). If set to learned(3), then it was found
                 via looking at the source IP address of a received
                 packet."
             ::= { docsDevCpeInetEntry 3 }

    docsDevCpeInetRowStatus OBJECT-TYPE
            SYNTAX RowStatus
            MAX-ACCESS read-create
            STATUS current
            DESCRIPTION
                "Standard object to manipulate rows. To create a row in
                 this table, one only needs to specify this object.
                 Management stations SHOULD use the create-and-go
                 mechanism for creating rows in this table."

```
             ::= { docsDevCpeInetEntry 4 }


     --
     -- Placeholder for notifications/traps.
     --

     -- erroneous, DO NOT USE docsDevNotification
     docsDevNotification OBJECT IDENTIFIER   ::= { docsDev 2 }
     -- erroneous, DO NOT USE docsDevNotification

     docsDevNotifications OBJECT IDENTIFIER    ::= { docsDev 0 }


     --
     -- RFC 2669 Conformance definitions
     --

     docsDevConformance  OBJECT IDENTIFIER   ::= { docsDev 3 }
     docsDevGroups       OBJECT IDENTIFIER   ::= { docsDevConformance 1 }
     docsDevCompliances  OBJECT IDENTIFIER   ::= { docsDevConformance 2 }

     docsDevBasicCompliance MODULE-COMPLIANCE
             STATUS   deprecated
             DESCRIPTION
                 "The RFC 2669 compliance statement for MCNS/DOCSIS
                  Cable Modems and Cable Modem Termination Systems."

     MODULE  -- docsDev

     -- conditionally mandatory groups

     GROUP docsDevBaseGroup
             DESCRIPTION
                 "Mandatory in Cable Modems, optional in Cable Modem
                  Termination Systems."

     GROUP docsDevEventGroup
             DESCRIPTION
                 "Mandatory in Cable Modems, optional in Cable Modem
                  Termination Systems."

     GROUP docsDevFilterGroup
             DESCRIPTION
                 "Mandatory in Cable Modems, optional in Cable Modem
                  Termination Systems."

     GROUP docsDevNmAccessGroup
```

                DESCRIPTION
                    "This group is only implemented in devices that do not
                    implement the SNMPv3 User Security Model. It SHOULD NOT
                    be implemented by devices that conform to SNMPv3.

                    For devices that do not implement SNMPv3 or later, this
                    group is Mandatory in Cable Modems and is optional
                    in Cable Modem Termination Systems."
        GROUP docsDevServerGroup
                DESCRIPTION
                    "This group is implemented only in Cable Modems, and is
                    not implemented in Cable Modem Termination Systems."

        GROUP docsDevSoftwareGroup
                DESCRIPTION
                    "This group is Mandatory in Cable Modems and optional in
                    Cable Modem Termination Systems."

        GROUP docsDevCpeGroup
                DESCRIPTION
                    "This group is Mandatory in Cable Modems, and is
                    not implemented in Cable Modem Termination Systems."

        OBJECT docsDevSTPControl
                MIN-ACCESS read-only
                DESCRIPTION
                    "It is compliant to implement this object as read-only.
                    Devices need only support noStFilterBpdu(2)."

        OBJECT docsDevNmAccessIp
                DESCRIPTION
                    "It is compliant to recognize the IP address
                    255.255.255.255 as referring to any NMS."

        OBJECT docsDevEvReporting
                 MIN-ACCESS read-only
                 DESCRIPTION
                    "It is compliant to implement this object as read-only.
                     Devices need only support local(0). An agent need not
                     enforce that trap or syslog logging be accompanied
                     by local(0) or localVolatile(3) logging."
                 ::= { docsDevCompliances 1 }

    docsDevBaseGroup OBJECT-GROUP
                OBJECTS {
                    docsDevRole,
                    docsDevDateTime,

```
                docsDevResetNow,
                docsDevSerialNumber,
                docsDevSTPControl
        }
        STATUS          current
        DESCRIPTION
            "A collection of objects providing device status and
            control."
        ::= { docsDevGroups 1 }

   docsDevNmAccessGroup OBJECT-GROUP
        OBJECTS {
                docsDevNmAccessIp,
                docsDevNmAccessIpMask,
                docsDevNmAccessCommunity,
                docsDevNmAccessControl,
                docsDevNmAccessInterfaces,
                docsDevNmAccessStatus
        }
        STATUS          deprecated
        DESCRIPTION
            "A collection of objects for controlling access to SNMP
            objects on cable devices.

            This group has been deprecated because all the
            objects have been deprecated in favor of SNMPv3 and
            Coexistence MIBs."
        ::= { docsDevGroups 2 }

   docsDevSoftwareGroup OBJECT-GROUP
        OBJECTS {
                docsDevSwServer,
                docsDevSwFilename,
                docsDevSwAdminStatus,
                docsDevSwOperStatus,
                docsDevSwCurrentVers
        }
        STATUS          deprecated
        DESCRIPTION
            "A collection of objects for controlling software
            downloads.

            This group has been deprecated and replaced by
            docsDevSoftwareGroupV2.  Object docsDevSwServer
            has been replaced by docsDevSwServerAddressType
            and docsDevSwServerAddress, and
            docsDevSwServerTransportProtocol has been added to
            support TFTP and HTTP firmware downloads."
```

```
         ::= { docsDevGroups 3 }

  docsDevServerGroup OBJECT-GROUP
         OBJECTS {
             docsDevServerBootState,
             docsDevServerDhcp,
             docsDevServerTime,
             docsDevServerTftp,
             docsDevServerConfigFile
         }
         STATUS      deprecated
         DESCRIPTION
             "A collection of objects providing status about server
              provisioning.

              This group has been deprecated and replaced by
              docsDevServerGroupV2.  The objects docsDevServerDhcp,
              docsDevServerTime, and docsDevServerTftp have
              been replaced by docsDevServerDhcpAddressType,
              docsDevServerDhcpAddress, docsDevServerTimeAddressType,
              docsDevServerTimeAddress,
              docsDevServerConfigTftpAddressType, and
              docsDevServerConfigTftpAddress."
         ::= { docsDevGroups 4 }

  docsDevEventGroup OBJECT-GROUP
         OBJECTS {
             docsDevEvControl,
             docsDevEvSyslog,
             docsDevEvThrottleAdminStatus,
             docsDevEvThrottleInhibited,
             docsDevEvThrottleThreshold,
             docsDevEvThrottleInterval,
             docsDevEvReporting,
             docsDevEvFirstTime,
             docsDevEvLastTime,
             docsDevEvCounts,
             docsDevEvLevel,
             docsDevEvId,
             docsDevEvText
         }
         STATUS      deprecated
         DESCRIPTION
             "A collection of objects used to control and monitor
              events.

              This group has been deprecated and replaced by
              docsDevEventGroupV2.  The object docsDevEvSyslog has
```

                    been replaced by docsDevEvSyslogAddressType and
                    docsDevEvSyslogAddress, and docsDevEvThrottleInhibited
                    has been replaced by
                    docsDevEvThrottleThresholdExceeded."
                ::= { docsDevGroups 5 }

    docsDevFilterGroup OBJECT-GROUP
            OBJECTS {
                docsDevFilterLLCUnmatchedAction,
                docsDevFilterIpDefault,
                docsDevFilterLLCStatus,
                docsDevFilterLLCIfIndex,
                docsDevFilterLLCProtocolType,
                docsDevFilterLLCProtocol,
                docsDevFilterLLCMatches,
                docsDevFilterIpControl,
                docsDevFilterIpIfIndex,
                docsDevFilterIpStatus,
                docsDevFilterIpDirection,
                docsDevFilterIpBroadcast,
                docsDevFilterIpSaddr,
                docsDevFilterIpSmask,
                docsDevFilterIpDaddr,
                docsDevFilterIpDmask,
                docsDevFilterIpProtocol,
                docsDevFilterIpSourcePortLow,
                docsDevFilterIpSourcePortHigh,
                docsDevFilterIpDestPortLow,
                docsDevFilterIpDestPortHigh,
                docsDevFilterIpMatches,
                docsDevFilterIpTos,
                docsDevFilterIpTosMask,
                docsDevFilterIpContinue,
                docsDevFilterIpPolicyId,
                docsDevFilterPolicyId,
                docsDevFilterPolicyStatus,
                docsDevFilterPolicyPtr,
                docsDevFilterTosStatus,
                docsDevFilterTosAndMask,
                docsDevFilterTosOrMask
            }
            STATUS   deprecated
            DESCRIPTION
                "A collection of objects to specify filters at the link
                 layer and IPv4 layer.

                 This group has been deprecated and replaced by various
                 groups from the DiffServ MIB."

```
        ::= { docsDevGroups 6 }

docsDevCpeGroup OBJECT-GROUP
        OBJECTS {
            docsDevCpeEnroll,
            docsDevCpeIpMax,
            docsDevCpeSource,
            docsDevCpeStatus
        }
        STATUS         deprecated
        DESCRIPTION
            "A collection of objects used to control the number
             and specific values of IPv4 addresses allowed for
             associated Customer Premises Equipment (CPE).

             This group has been deprecated and replaced by
             docsDevInetCpeGroup.  The object docsDevCpeSource has
             been replaced by docsDevCpeInetSource, and
             docsDevCpeStatus has been replaced by
             docsDevCpeInetRowStatus."
        ::= { docsDevGroups 7 }

--
-- RFC 4639 Conformance definitions
--

docsDevGroupsV2       OBJECT IDENTIFIER  ::= { docsDevConformance 3 }
docsDevCompliancesV2  OBJECT IDENTIFIER  ::= { docsDevConformance 4 }

docsDevCmCompliance MODULE-COMPLIANCE
        STATUS   current
        DESCRIPTION
            "The compliance statement for DOCSIS Cable Modems.

             This compliance statement applies to implementations
             of DOCSIS versions that are not IPv6 capable."

MODULE DIFFSERV-MIB -- RFC 3289

MANDATORY-GROUPS {
        diffServMIBDataPathGroup,
        diffServMIBClfrGroup,
        diffServMIBClfrElementGroup,
        diffServMIBMultiFieldClfrGroup,
        diffServMIBActionGroup,
        diffServMIBDscpMarkActGroup,
        diffServMIBCounterGroup,
        diffServMIBAlgDropGroup
```

```
                }

    OBJECT diffServDataPathStatus  -- same as RFC 3289
            SYNTAX RowStatus { active(1) }
            WRITE-SYNTAX RowStatus { createAndGo(4), destroy(6) }
            DESCRIPTION
                "Support for createAndWait and notInService is not
                 required."

    OBJECT diffServClfrStatus  -- same as RFC 3289
            SYNTAX RowStatus { active(1) }
            WRITE-SYNTAX RowStatus { createAndGo(4), destroy(6) }
            DESCRIPTION
                "Support for createAndWait and notInService is not
                 required."

    OBJECT diffServClfrElementStatus  -- same as RFC 3289
            SYNTAX RowStatus { active(1) }
            WRITE-SYNTAX RowStatus { createAndGo(4), destroy(6) }
            DESCRIPTION
                "Support for createAndWait and notInService is not
                 required."

    OBJECT diffServMultiFieldClfrAddrType
            SYNTAX  InetAddressType { ipv4(1) }
            DESCRIPTION
                "An implementation is only required to support IPv4
                 addresses."

    OBJECT diffServMultiFieldClfrSrcAddr
            SYNTAX  InetAddress (SIZE(4))
            DESCRIPTION
                "An implementation is only required to support IPv4
                 addresses."

    OBJECT diffServMultiFieldClfrDstAddr
            SYNTAX  InetAddress (SIZE(4))
            DESCRIPTION
                "An implementation is only required to support IPv4
                 addresses."

    OBJECT diffServAlgDropStatus  -- same as RFC 3289
            SYNTAX RowStatus { active(1) }
            WRITE-SYNTAX RowStatus { createAndGo(4), destroy(6) }
            DESCRIPTION
                "Support for createAndWait and notInService is not
                 required."
```

```
   OBJECT diffServDataPathStorage
           SYNTAX StorageType { volatile(2) }
           DESCRIPTION
               "An implementation is only required to support
                volatile storage."

   OBJECT diffServClfrStorage
           SYNTAX StorageType { volatile(2) }
           DESCRIPTION
               "An implementation is only required to support
                volatile storage."

   OBJECT diffServClfrElementStorage
           SYNTAX StorageType { volatile(2) }
           DESCRIPTION
               "An implementation is only required to support
                volatile storage."

   OBJECT diffServMultiFieldClfrStorage
           SYNTAX StorageType { volatile(2) }
           DESCRIPTION
               "An implementation is only required to support
                volatile storage."

   OBJECT diffServActionStorage
           SYNTAX StorageType { volatile(2) }
           DESCRIPTION
               "An implementation is only required to support
                volatile storage."

   OBJECT diffServCountActStorage
           SYNTAX StorageType { volatile(2) }
           DESCRIPTION
               "An implementation is only required to support
                volatile storage."

   OBJECT diffServAlgDropStorage
           SYNTAX StorageType { volatile(2) }
           DESCRIPTION
               "An implementation is only required to support
                volatile storage."

   OBJECT diffServAlgDropType
           SYNTAX  INTEGER { alwaysDrop(5) }
           DESCRIPTION
               "This object is only used to provide packet
                filtering.  Implementations need not support other
                values of this enumeration."
```

```
MODULE  -- docsDev

MANDATORY-GROUPS {
            docsDevBaseGroup,
            docsDevBaseIgmpGroup,
            docsDevBaseMaxCpeGroup,
            docsDevSoftwareGroupV2,
            docsDevServerGroupV2,
            docsDevEventGroupV2,
            docsDevFilterLLCGroup
            }

-- conditionally mandatory groups

GROUP docsDevInetCpeGroup
        DESCRIPTION
            "This group is optional in Cable Modems."

OBJECT docsDevDateTime
        MIN-ACCESS read-only
        DESCRIPTION
            "It is compliant to implement this object as read-only."

OBJECT docsDevSTPControl
        SYNTAX INTEGER { noStFilterBpdu(2) }
        MIN-ACCESS read-only
        DESCRIPTION
            "It is compliant to implement this object as read-only.
             Devices need only support noStFilterBpdu(2)."

OBJECT docsDevIgmpModeControl
        SYNTAX INTEGER { passive(1) }
        MIN-ACCESS read-only
        DESCRIPTION
            "It is compliant to implement this object as read-only.
             Devices need only support passive(1)."

OBJECT docsDevSwServerAddressType
        SYNTAX InetAddressType { ipv4(1) }
        DESCRIPTION
            "An implementation is only required to support IPv4
             addresses."

OBJECT docsDevSwServerAddress
        SYNTAX  InetAddress (SIZE(4))
        DESCRIPTION
            "An implementation is only required to support IPv4
             addresses."
```

```
    OBJECT  docsDevServerDhcpAddressType
            SYNTAX InetAddressType { ipv4(1) }
            DESCRIPTION
                "An implementation is only required to support IPv4
                 addresses."

    OBJECT  docsDevServerDhcpAddress
            SYNTAX  InetAddress (SIZE(4))
            DESCRIPTION
                "An implementation is only required to support IPv4
                 addresses."

    OBJECT  docsDevServerTimeAddressType
            SYNTAX InetAddressType { ipv4(1) }
            DESCRIPTION
                "An implementation is only required to support IPv4
                 addresses."

    OBJECT  docsDevServerTimeAddress
            SYNTAX  InetAddress (SIZE(4))
            DESCRIPTION
                "An implementation is only required to support IPv4
                 addresses."

    OBJECT  docsDevServerConfigTftpAddressType
            SYNTAX InetAddressType { ipv4(1) }
            DESCRIPTION
                "An implementation is only required to support IPv4
                 addresses."

    OBJECT  docsDevServerConfigTftpAddress
            SYNTAX  InetAddress (SIZE(4))
            DESCRIPTION
                "An implementation is only required to support IPv4
                 addresses."

    OBJECT  docsDevEvReporting
            MIN-ACCESS read-only
            DESCRIPTION
                "It is compliant to implement this object as read-only.
                 Devices need only support local(0)."

    OBJECT  docsDevEvSyslogAddressType
            SYNTAX InetAddressType { ipv4(1) }
            DESCRIPTION
                "An implementation is only required to support IPv4
                 addresses."
```

```
OBJECT docsDevEvSyslogAddress
        SYNTAX  InetAddress (SIZE(4))
        DESCRIPTION
            "An implementation is only required to support IPv4
             addresses."

OBJECT docsDevSwServerTransportProtocol
        SYNTAX INTEGER { tftp(1) }
        DESCRIPTION
            "An implementation is only required to support TFTP
             software image downloads."

        ::= { docsDevCompliancesV2 1 }

docsDevCmtsCompliance MODULE-COMPLIANCE
        STATUS  current
        DESCRIPTION
            "The compliance statement for DOCSIS Cable Modem
             Termination Systems.

             This compliance statement applies to implementations
             of DOCSIS versions that are not IPv6 capable."

    MODULE  -- docsDev

    -- conditionally mandatory groups

    GROUP docsDevBaseGroup
            DESCRIPTION
                "Optional in Cable Modem Termination Systems."

    GROUP docsDevBaseIgmpGroup
            DESCRIPTION
                "Optional in Cable Modem Termination Systems."

    GROUP docsDevBaseMaxCpeGroup
            DESCRIPTION
                "This group MUST NOT be implemented in Cable Modem
                 Termination Systems."

    GROUP docsDevSoftwareGroupV2
            DESCRIPTION
                "Optional in Cable Modem Termination Systems."

    GROUP docsDevServerGroupV2
            DESCRIPTION
                "This group MUST NOT be implemented in Cable Modem
                 Termination Systems."
```

         GROUP docsDevEventGroupV2
                 DESCRIPTION
                     "Optional in Cable Modem Termination Systems."

         GROUP docsDevFilterLLCGroup
                 DESCRIPTION
                     "This group MUST NOT be implemented in Cable Modem
                      Termination Systems. See the Subscriber Management
                      MIB for similar CMTS capability."

         GROUP docsDevInetCpeGroup
                 DESCRIPTION
                     "This group MUST NOT be implemented in Cable Modem
                      Termination Systems. See the Subscriber Management
                      MIB for similar CMTS capability."

         OBJECT docsDevDateTime
                 MIN-ACCESS read-only
                 DESCRIPTION
                     "It is compliant to implement this object as read-only."

         OBJECT docsDevSTPControl
                 SYNTAX INTEGER { noStFilterBpdu(2) }
                 MIN-ACCESS read-only
                 DESCRIPTION
                     "It is compliant to implement this object as read-only.
                      Devices need only support noStFilterBpdu(2)."

         OBJECT docsDevIgmpModeControl
                 SYNTAX INTEGER { passive(1) }
                 MIN-ACCESS read-only
                 DESCRIPTION
                     "It is compliant to implement this object as read-only.
                      Devices need only support passive(1)."

         OBJECT docsDevSwServerAddressType
                 SYNTAX InetAddressType { ipv4(1) }
                 DESCRIPTION
                     "An implementation is only required to support IPv4
                      addresses."

         OBJECT docsDevSwServerAddress
                 SYNTAX InetAddress (SIZE(4))
                 DESCRIPTION
                     "An implementation is only required to support IPv4
                      addresses."

         OBJECT docsDevEvReporting

```
            MIN-ACCESS read-only
            DESCRIPTION
                "It is compliant to implement this object as read-only.
                 Devices need only support local(0)."

    OBJECT docsDevEvSyslogAddressType
            SYNTAX InetAddressType { ipv4(1) }
            DESCRIPTION
                "An implementation is only required to support IPv4
                 addresses."

    OBJECT docsDevEvSyslogAddress
            SYNTAX  InetAddress (SIZE(4))
            DESCRIPTION
                "An implementation is only required to support IPv4
                 addresses."

    OBJECT docsDevSwServerTransportProtocol
            SYNTAX INTEGER { tftp(1) }
            DESCRIPTION
                "An implementation is only required to support TFTP
                 software image downloads."

            ::= { docsDevCompliancesV2 2 }

    docsDevBaseIgmpGroup OBJECT-GROUP
            OBJECTS {
                docsDevIgmpModeControl
            }
            STATUS        current
            DESCRIPTION
                "An object providing cable device IGMP status and
                 control."
            ::= { docsDevGroupsV2 1 }

    docsDevBaseMaxCpeGroup OBJECT-GROUP
            OBJECTS {
                docsDevMaxCpe
            }
            STATUS        current
            DESCRIPTION
                "An object providing management of the maximum number of
                 CPEs permitted access through a cable modem."
            ::= { docsDevGroupsV2 2 }

    docsDevNmAccessExtGroup OBJECT-GROUP
            OBJECTS {
                docsDevNmAccessTrapVersion
```

```
            }
            STATUS        deprecated
            DESCRIPTION
                "An object, in addition to the objects in
                 docsDevNmAccessGroup, for controlling access to
                 SNMP objects on cable devices.

                 This group is included in this MIB due to existing
                 implementations of docsDevNmAccessTrapVersion in
                 DOCSIS cable modems.

                 This group has been deprecated because the object has
                 been deprecated in favor of SNMPv3 and Coexistence
                 MIBs."
            ::= { docsDevGroupsV2 3 }

    docsDevSoftwareGroupV2 OBJECT-GROUP
            OBJECTS {
                docsDevSwFilename,
                docsDevSwAdminStatus,
                docsDevSwOperStatus,
                docsDevSwCurrentVers,
                docsDevSwServerAddressType,
                docsDevSwServerAddress,
                docsDevSwServerTransportProtocol
            }
            STATUS        current
            DESCRIPTION
                "A collection of objects for controlling software
                 downloads.  This group replaces docsDevSoftwareGroup."
            ::= { docsDevGroupsV2 4 }

    docsDevServerGroupV2 OBJECT-GROUP
            OBJECTS {
                docsDevServerBootState,
                docsDevServerDhcpAddressType,
                docsDevServerDhcpAddress,
                docsDevServerTimeAddressType,
                docsDevServerTimeAddress,
                docsDevServerConfigTftpAddressType,
                docsDevServerConfigTftpAddress,
                docsDevServerConfigFile
            }
            STATUS        current
            DESCRIPTION
                "A collection of objects providing status about server
                 provisioning.  This group replaces docsDevServerGroup."
            ::= { docsDevGroupsV2 5 }
```

```
docsDevEventGroupV2 OBJECT-GROUP
        OBJECTS {
            docsDevEvControl,
            docsDevEvThrottleAdminStatus,
            docsDevEvThrottleThreshold,
            docsDevEvThrottleInterval,
            docsDevEvReporting,
            docsDevEvFirstTime,
            docsDevEvLastTime,
            docsDevEvCounts,
            docsDevEvLevel,
            docsDevEvId,
            docsDevEvText,
            docsDevEvSyslogAddressType,
            docsDevEvSyslogAddress,
            docsDevEvThrottleThresholdExceeded
        }
        STATUS        current
        DESCRIPTION
            "A collection of objects used to control and monitor
             events.  This group replaces docsDevEventGroup.
             The event reporting mechanism, and more specifically
             docsDevEvReporting, can be used to take advantage of
             the event reporting features of RFC3413 and RFC3014."
        ::= { docsDevGroupsV2 6 }

docsDevFilterLLCGroup OBJECT-GROUP
        OBJECTS {
            docsDevFilterLLCUnmatchedAction,
            docsDevFilterLLCStatus,
            docsDevFilterLLCIfIndex,
            docsDevFilterLLCProtocolType,
            docsDevFilterLLCProtocol,
            docsDevFilterLLCMatches
        }
        STATUS        current
        DESCRIPTION
            "A collection of objects to specify link layer filters."
        ::= { docsDevGroupsV2 7 }

docsDevInetCpeGroup OBJECT-GROUP
        OBJECTS {
            docsDevCpeEnroll,
            docsDevCpeIpMax,
            docsDevCpeInetSource,
            docsDevCpeInetRowStatus
        }
        STATUS        current
```

```
            DESCRIPTION
                "A collection of objects used to control the number
                 and specific values of Internet (e.g., IPv4 and IPv6)
                 addresses allowed for associated Customer Premises
                 Equipment (CPE)."
            ::= { docsDevGroupsV2 8 }

    END
```

5. Acknowledgements

This document is a production of the IPCDN Working Group and is a
revision of RFC 2669, "Cable Device Management Information Base for
DOCSIS-Compliant Cable Modems and Cable Modem Termination Systems"
[RFC2669]. Mike St. Johns and Guenter Roeck served well as the
editors of previous versions of this MIB module.

The editor specifically wishes to thank Howard Abramson, Eduardo
Cardona, Andre Lejeune, Kevin Marez, Jean-Francois Mule, Greg
Nakanishi, Pak Siripunkaw, Boris Tsekinovski, Randy Presuhn, Bert
Wijnen, and Bill Yost for their contributions to this document.

5.1. Revision Descriptions

This document contains the following revisions over RFC 2669:

o All IPv4 address objects were either deprecated and replaced or
   mirrored with IPv6 objects, where appropriate, following the
   guidelines of RFC 4001 [RFC4001]. In particular,
   docsDevCpeInetTable was added, and the docsDevFilterGroup objects
   were deprecated in favor of the DiffServ MIB.

o Objects that were obviated by SNMPv3 and the SNMP Coexistence MIBs
   have been deprecated; e.g., docsDevNmAccessTable.

o A new object, docsDevIgmpModeControl, has been added to control
   passive versus active IGMP modem operation.

o A new object, docsDevMaxCpe, has been added to report the maximum
   number of CPEs granted network access across the CM.

o A new object, docsDevSwServerTransportProtocol, has been added to
   docsDevSoftware, and other object DESCRIPTIONs have been modified,
   to enable the use of either TFTP or HTTP for software downloads to
   the device.

   o A new object, docsDevEvThrottleThresholdExceeded, has been added
      to replace docsDevEvThrottleInhibited for simplification of event
      threshold management.

   o The docsDevEvReporting object has been modified to enable local
      logging to the internal volatile log, and not to the internal
      non-volatile log.

   o Minor updates to the description text have been made to a number
      of objects to clarify their meaning.

   o The compliance statements were updated to reflect current
      requirements (including making the docsDevCpe objects optional)
      and split between CM and CMTS devices.

   o Text was added to indicate support of the SNMP Notification MIB
      [RFC3413] and Notification Log MIB [RFC3014] modules.

6. Security Considerations

   This MIB module relates to a system that will provide metropolitan
   public internet access. As such, improper manipulation of the
   objects represented by this MIB module may result in denial of
   service to a large number of end-users. In addition, manipulation of
   docsDevNmAccessTable, docsDevFilterLLCTable, docsDevFilterIpTable,
   docsDevFilterInetTable, and the elements of the docsDevCpe and
   docsDevCpeInetTable groups may allow an end-user to increase his or
   her service levels, spoof his or her IP addresses, change the
   permitted management stations, or affect other end-users in either a
   positive or negative manner.

   It is recommended that the implementors prevent the "tiny fragment"
   and "overlapping fragment" attacks for the IP filtering tables in
   this MIB module, as discussed in [RFC1858] and [RFC3128]. Prevention
   of these attacks can be implemented with the following rules, when
   TCP source and/or destination port filtering is enabled:

   o Admit all packets with fragment offset >= 2.

   o Discard all packets with fragment offset = 1, or with fragment
      offset = 0 AND fragment payload length < 16.

   o Apply filtering rules to all packets with fragment offset = 0.

   This MIB module does not affect confidentiality of services on a
   cable modem system. [BPI] and [BPIPLUS] specify the implementation
   of the DOCSIS Baseline Privacy and Baseline Privacy Plus mechanisms
   for data transmission confidentiality.

There are a number of management objects defined in this 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. These are the tables and objects and their
sensitivity/vulnerability:

o The use of docsDevNmAccessTable to specify management stations is
   considered only limited protection and does not protect against
   attacks that spoof the management station's IP address. The use
   of stronger mechanisms, such as SNMPv3 security, should be
   considered, where possible. Specifically, SNMPv3 USM [RFC3414]
   and VACM [RFC3415] MUST be used with any v3 agent that implements
   this MIB module.

o The CM may have its software changed by the actions of the
   management system using a combination of the following objects:
   docsDevSwServer, docsDevSwFilename, docsDevSwAdminStatus,
   docsDevSwServerAddressType, docsDevSwServerAddress, and
   docsDevSwServerTransportProtocol. An improper software download
   may result in substantial vulnerabilities and the loss of the
   ability of the management system to control the cable modem. A
   cable device SHOULD implement the code verification mechanisms of
   [BPIPLUS] to verify the source and integrity of downloaded
   software images.

o The device may be reset by setting docsDevResetNow = true(1).
   This causes the device to reload its configuration files, as well
   as to eliminate all previous non-persistent network management
   settings. As such, this may provide a vector for attacking the
   system.

o Setting docsDevEvThrottleAdminStatus = unconstrained(1) (which is
   also the DEFVAL) may cause flooding of traps, which can disrupt
   network service. Additionally, docsDevThrottleThreshold and
   docsDevThrottleInterval could also be set to high values that may
   cause a disruption in service.

o Setting docsDevDateTime to an arbitrary (incorrect) value would
   merely cause the device to record incorrect timestamps on many
   events/actions that rely on this object for reporting.

o Setting docsDevEvControl to resetLog(1) will delete any event log
   history and could potentially impact debugging/troubleshooting
   efforts.

o Setting docsDevEvSyslog.

o Setting docsDevEvReporting to enable syslog reporting, along with
   a redirect of the syslog server could allow access to sensitive
   information on network devices. Modifying docsDevEvSyslog,
   docsDevEvSyslogAddressType, or docsDevEvSyslogAddress could allow
   a redirect of sensitive information.

o Setting docsDevFilterLLCnmatchedAction or docsDevFilterIpDefault
   could cause significant changes to default traffic filtering on a
   device.

o Setting docsDevCpeEnroll to any(2) could cause the
   docsDevFilterCPETable to be populated, which may not be the
   intended functionality.

o Setting docsDevCpeIpMax to a value other than that intended by the
   MSO may allow a user to provision more devices than the MSO would
   like.

o Setting values in the docsDevNmAccess table can potentially
   introduce a mechanism for users to use a local NMS device and
   manipulate other settings in the CM or CMTS.

o Setting values in the docsDevFilterLLC and docsDevFilterIP tables
   can allow or deny access to certain devices that the MSO does not
   want.

o Setting docsDevCpeStatus and docsDevCpeInetRowStatus may allow
   users to provision more devices than were intended by the MSO, or
   to provision different ones.

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

o Rows from docsDevNmAccessTable may provide sufficient information
   for attackers to spoof management stations that have management
   access to the device.

o The docsDevSwCurrentVers object may provide hints as to the
   software vulnerabilities of the cable device.

o The docsDevFilterLLCTable and docsDevFilterLLCTable may provide
   clues for attacking the cable device and other subscriber devices.

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

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

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

7. IANA Considerations

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

```
Descriptor OBJECT IDENTIFIER value
---------- ------------------------
docsDevMIB { mib-2 69 }
```

8. References

8.1. Normative References

[BPI]   SCTE Data Standards Subcommittee, "Data-Over-Cable
                 Service Interface Specifications: DOCSIS 1.0 Baseline
                 Privacy Interface Specification SCTE 22-2 2002", 2002,
                 <http://www.scte.org/standards/>.

[BPIPLUS]   CableLabs, "Data-Over-Cable Service Interface
                 Specifications: Baseline Privacy Plus Interface
                 Specification CM-SP-BPI+_I12-050812", August 2005,
                 <http://www.cablemodem.com/specifications/>,
                 <http://www.cablelabs.com/specifications/archives/>.

[ITU-T_J.112]   ITU-T Recommendation J.112 (3/98), "Transmission
                 Systems for Interactive Cable Television Services,
                 J.112, International Telecommunications Union", March
                 1998, <http://www.itu.int/ITU-T/studygroups/com09/>.

[MTA-PROV]   CableLabs, "PacketCable(TM) 1.5 Specification: MTA
                 Device Provisioning PKT-SP-PROV1.5-I02-050812", August
                 2005, <http://www.packetcable.com/specifications/>,
                 <http://www.cablelabs.com/specifications/archives/>.

[OSSI1.0]   SCTE Data Standards Subcommittee, "Data-Over-Cable
                 Service Interface Specification: DOCSIS 1.0 Operations
                 Support System Interface (OSSI), SCTE 22-3 2002", 2002,
                 <http://www.scte.org/standards/>.

[OSSI1.1]   SCTE Data Standards Subcommittee, "DOCSIS 1.1 Part 3:
                 Operations Support System Interface ANSI/SCTE 23-3
                 2005", 2005, <http://www.scte.org/standards/>.

[OSSI2.0]   CableLabs, "Data-Over-Cable Service Interface
                 Specifications: Operations Support System Interface
                 Specification SP-OSSIv2.0-I09-050812", August 2005,
                 <http://www.cablemodem.com/specifications/>,
                 <http://www.cablelabs.com/specifications/archives/>.

[RFC1350]   Sollins, K., "The TFTP Protocol (Revision 2)", STD 33,
                 RFC 1350, July 1992.

[RFC4502]   Waldbusser, S., "Remote Network Monitoring Management
                 Information Base Version 2", RFC 4502, May 2006.

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

    [RFC2578]   McCloghrie, K., Perkins, D., Schoenwaelder J., Case,
                    J., Rose, M. and S. Waldbusser, "Structure of
                    Management Information Version 2 (SMIv2)", STD 58, RFC
                    2578, April 1999.

    [RFC2579]   McCloghrie, K., Perkins, D., Schoenwaelder, J., Case,
                    J., Rose, M. and S. Waldbusser , "Textual Conventions
                    for SMIv2", STD 58, RFC 2579, April 1999.

    [RFC2580]   McCloghrie, K., Perkins, D., Schoenwaelder, J., Case,
                    J., Rose, M. and S. Waldbusser, "Conformance Statements
                    for SMIv2", STD 58, RFC 2580, April 1999.

    [RFC2616]   Fielding, R., Gettys, J., Mogul, J., Frystyk, H.,
                    Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext
                    Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

    [RFC2669]   St. Johns, M., "DOCSIS Cable Device MIB Cable Device
                    Management Information Base for DOCSIS compliant Cable
                    Modems and Cable Modem Termination Systems", RFC 2669,
                    August 1999.

    [RFC2863]   McCloghrie, K. and F. Kastenholz, "The Interfaces Group
                    MIB", RFC 2863, June 2000.

    [RFC3014]   Kavasseri, R., "Notification Log MIB", RFC 3014,
                    November 2000.

    [RFC3289]   Baker, F., Chan, K., and A. Smith, "Management
                    Information Base for the Differentiated Services
                    Architecture", RFC 3289, May 2002.

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

    [RFC3413]   Levi, D., Meyer, P., and B. Stewart, "Simple Network
                    Management Protocol (SNMP) Applications", STD 62, RFC
                    3413, December 2002.

    [RFC3414]   Blumenthal, U. and B. Wijnen, "User-based Security
                    Model (USM) for version 3 of the Simple Network
                    Management Protocol (SNMPv3)", STD 62, RFC 3414,
                    December 2002.

   [RFC3415]   Wijnen, B., Presuhn, R., and K. McCloghrie, "View-based
                  Access Control Model (VACM) for the Simple Network
                  Management Protocol (SNMP)", STD 62, RFC 3415, December
                  2002.

   [RFC3418]   Presuhn, R., "Management Information Base (MIB) for the
                  Simple Network Management Protocol (SNMP)", STD 62, RFC
                  3418, December 2002.

   [RFC3584]   Frye, R., Levi, D., Routhier, S., and B. Wijnen,
                  "Coexistence between Version 1, Version 2, and Version
                  3 of the Internet-standard Network Management
                  Framework", BCP 74, RFC 3584, August 2003.

   [RFC868]   Postel, J. and K. Harrenstien, "Time Protocol", STD 26,
                  RFC 868, May 1983.

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

   [RFI1.0]   SCTE Data Standards Subcommittee, "Data-Over-Cable
                  Service Interface Specifications: DOCSIS 1.0 Radio
                  Frequency Interface Specification SCTE 22-1 2002",
                  2002, <http://www.scte.org/standards/>.

   [RFI1.1]   SCTE Data Standards Subcommittee, "DOCSIS 1.1 Part 1:
                  Radio Frequency Interface ANSI/SCTE 23-1 2005", 2005,
                  <http://www.scte.org/standards/>.

   [RFI2.0]   CableLabs, "Data-Over-Cable Service Interface
                  Specifications: Radio Frequency Interface Specification
                  SP-RFI2.0-I11-060602", June 2006,
                  <http://www.cablemodem.com/specifications/>,
                  <http://www.cablelabs.com/specifications/archives/>.

8.2. Informative References

   [RFC1858]   Ziemba, G., Reed, D., and P. Traina, "Security
                  Considerations r IP Fragment Filtering", RFC 1858,
                  October 1995.

   [RFC1945]   Berners-Lee, T., Fielding, R., and H. Frystyk,
                  "Hypertext Traner Protocol -- HTTP/1.0", RFC 1945, May
                  1996.

   [RFC3128]   Miller, I., "Protection Against a Variant of the Tiny
                  Fragment Attack (RFC 1858)", RFC 3128, June 2001.

   [RFC3164]   Lonvick, C., "The BSD Syslog Protocol", RFC 3164,
                August 2001.

   [RFC3617]   Lear, E., "Uniform Resource Identifier (URI) Scheme and
                Applicbility Statement for the Trivial File Transfer
                Protocol (TFTP)", RFC 3617, October 2003.

   [RFC4547]   Ahmad, A. and G. Nakanishi, "Event Notification
                Management Information Base for Data over Cable Service
                Interface Specifications (DOCSIS) Compliant Cable
                Modems and Cable Modem Termination Systems", RFC 4547,
                June 2006.

   [RFC1224]   Steinberg, L., "Techniques for managing asynchronously
                generated alerts", RFC 1224, May 1991.

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

   [RFC4036]   Sawyer, W., "Management Information Base for Data Over
                Cable Service Interface Specification (DOCSIS) Cable
                Modem Termination Systems for Subscriber Management",
                RFC 4036, April 2005.

   [RFC4323]   Patrick, M. and W. Murwin, "Data Over Cable System
                Interface Specification Quality of Service Management
                Information Base (DOCSIS-QoS MIB)", RFC 4323, January
                2006.

   [MULPI3.0]   CableLabs, "Data-Over-Cable Service Interface
                Specifications: DOCSIS 3.0 MAC and Upper Layer
                Protocols Interface Specification CM-SP-MULPIv3.0-I01-
                060804", August 2006,
                <http://www.cablemodem.com/specifications/>,
                <http://www.cablelabs.com/specifications/archives/>.

Authors' Addresses

   Richard Woundy
   Comcast Cable
   27 Industrial Avenue
   Chelmsford, MA 01824
   USA

   Phone: +1 978 244 4010
   EMail: richard_woundy@cable.comcast.com

   Kevin Marez
   Motorola Corporation
   6450 Sequence Drive
   San Diego, CA 92121
   USA

   Phone: +1 858 404 3785
   EMail: kevin.marez@motorola.com

Full Copyright Statement

Intellectual Property

Acknowledgement