

Network Working Group
Request for Comments: 5428
Category: Standards Track

S. Channabasappa
CableLabs
W. De Ketelaere
tComLabs
E. Nechamkin
Broadcom Corp.
April 2009

Management Event Management Information Base (MIB)
for PacketCable- and IPCablecom-Compliant Devices

Status of This Memo

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

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (<http://trustee.ietf.org/license-info>). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

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 events that can be generated by PacketCable- and IPCablecom-compliant Multimedia Terminal Adapter devices.

Table of Contents

1. The Internet-Standard Management Framework	2
2. Introduction	2
3. Terminology	3
3.1. PacketCable	3
3.2. IPCablecom	3
3.3. MTA	4
3.4. Endpoint	4
3.5. MSO	4
3.6. UDP	4
4. Overview	4
4.1. Structure of the MIB	5
4.2. pktcEventControl	6
4.3. pktcEventThrottle	6
4.4. pktcEventStatus	7
4.5. pktcEvent	7
4.6. pktcEventLog	7
4.7. pktcEventNotifications	7
5. Relationship to Other MIB Modules	7
5.1. MIB Modules Required for IMPORTS	7
6. Definitions	8
7. IANA Considerations	32
8. Security Considerations	32
9. Acknowledgments	34
10. Normative References	35
11. Informative References	36

1. The Internet-Standard Management Framework

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

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

2. Introduction

A Multimedia Terminal Adapter (MTA) is used to deliver broadband Internet, data, and/or voice access jointly with telephony service to a subscriber's or customer's premises using a cable network

infrastructure. An MTA is normally installed at the subscriber's or customer's premises and is coupled to a multiple system operator (MSO) using a hybrid fiber coax (HFC) access network.

An MTA is provisioned by the MSO for broadband Internet, data, and/or voice service. For more information on MTA provisioning, refer to [PKT-SP-PROV] and [RFC4682]. MTA devices include one or more endpoints (e.g., telephone ports), which receive call signaling information to establish ring cadence, and codecs, which provide telephony service.

For more information on call signaling refer to, [PKT-SP-MGCP] and [RFC3435].

For more information on codecs, refer to [PKT-SP-CODEC].

Given the complexity of such systems, it is important that a suitable event management mechanism be defined to allow for effective management. This MIB module provides objects suitable for generation and management of events on the MTA.

3. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

The terms "MIB module" and "information module" are used interchangeably in this memo. As used here, both terms refer to any of the three types of information modules defined in Section 3 of RFC 2578 [RFC2578]. Some of the terms used in this memo are defined below. Some additional terms are also defined in the PacketCable(TM) Management Event Mechanism Specification [PKT-SP-MEM1.5] and the PacketCable MTA Device Provisioning Specification [PKT-SP-PROV].

3.1. PacketCable

PacketCable is a CableLabs-led initiative that is aimed at developing interoperable interface specifications for delivering advanced, real-time multimedia services over two-way cable plants.

3.2. IPCablecom

IPCablecom is an ITU Telecommunication Standardization Sector (ITU-T) project that includes architecture and a series of recommendations that enable the delivery of real-time services over the cable television networks using cable modems.

3.3. MTA

A Multimedia Terminal Adapter (MTA) is a PacketCable- or IPCablecom-compliant device providing telephony services over a cable or hybrid system used to deliver video signals to a community. It contains an interface to endpoints, a network interface, codecs, and all signaling and encapsulation functions required for Voice over IP transport, call signaling, and Quality of Service signaling. An MTA can be an embedded or standalone device. An Embedded MTA (E-MTA) is an MTA device containing an embedded Data Over Cable Service Interface Specifications (DOCSIS) cable modem. A Standalone MTA (S-MTA) is an MTA device separated from the DOCSIS cable modem by a non-DOCSIS Media Access Control (MAC) interface (e.g., Ethernet, USB).

3.4. Endpoint

An endpoint or MTA endpoint is a standard RJ-11 telephony physical port located on the MTA and used for attaching the telephone device to the MTA.

3.5. MSO

A Multi-System Operator is a cable company that operates many head-end locations in several cities.

3.6. UDP

A User Datagram Protocol is a connectionless protocol built upon Internet Protocol (IP), as per RFC 768 [RFC768].

4. Overview

PacketCable, European Telecommunications Standards Institute (ETSI), and International Telecommunication Union Telecommunication Standardization Sector (ITU-T) IPCablecom-compliant Multimedia Terminal Adaptors (MTAs) are required to generate management events upon the occurrence of certain operational conditions (for instance, "AC power failure, MTA operational on battery power"). The complete set of conditions and the corresponding management events to be generated are specified in [PKT-SP-MEM1.5] (PacketCable), [ETSITS101909-22] (ETSI), and [ITU-T-J176] (ITU-T). In addition, the MTA manufacturer is allowed to specify vendor-specific management events. For example, vendor XYZ can specify "Memory read error, terminating process, code: XYZ123".

When management events are generated, they can either be stored in a local log on the MTA or transmitted using two possible mechanisms: SNMP or syslog. This choice between storing and transmitting is required to be configurable and manageable by the management station for each management event (default values can be provided when the events are defined). This document proposes a MIB that can provide for configuration and management of such management events. A means to log the events is provided within the specified MIB module. For syslog as a transport, the necessary information (format, transport, etc.) is also specified. For SNMP as a transport, the MIB objects specified in the SNMP-TARGET-MIB and SNMP-NOTIFICATION-MIB as utilized, is specified in [RFC3413].

Further, each management event can be uniquely identified using the 'Organization ID' and 'Event ID'. The 'Organization ID' is the private enterprise number of the organization specifying the event (e.g., 4491 for CableLabs) and a unique identifier that identifies the event. The 'Event ID' is an identifier that uniquely identifies the event within the 'Organization ID' space. This document does not specify any management events. It only provides a mechanism to manage the storage and transmission of events.

The EVENT MIB module specified in this document is intended to update the EVENT MIB modules from which it is partly derived:

- the PacketCable 1.5 Management Event MIB Specification [PKT-SP-EVEMIB1.5] and
- the ITU-T IPCablecom management event mechanism MIB requirements [ITU-T-J176].

Several normative and informative references are used to help define Management Event MIB objects. As a convention, wherever the requirements are equivalent at the time of the writing, the PacketCable reference is used. However, MTA implementations MUST refer to the corresponding specifications to ensure compliance.

4.1. Structure of the MIB

The Management Event MIB module is identified by `pktcIetfEventMib` and is structured into the following sub-trees:

- `pktcEventControl` specifies the management information pertinent to control of the device's event generation capabilities.
- `pktcEventThrottle` specifies the management information pertinent to throttling the transmission of management events using syslog or SNMP.

- pktcEventStatus specifies the management information for the device to report status information related to the generated events.
- pktcEvents specifies the management information for the device to list all the events it is capable of generating.
- pktcEventLog specifies the management information for the device to store the generated events.
- pktcEventNotifications specifies the management information that defines the SNMP trap and inform messages.

4.2. pktcEventControl

The group of objects in this sub-tree provide for three important controls: ability to reset the event logs and event descriptions, syslog configuration, and event classes.

Some highlights are as follows:

pktcEventReset - this MIB object allows a management station to reset the event logs, the event descriptions, or both.

pktcEventSyslog - this group of MIB objects allows the management station to provide information for transmission of events to a syslog server, such as message formats and transport protocols.

pktcEventClassTable - this MIB table allows for MTAs to classify the management events into different categories, termed 'event classes'. It then allows for common operations to be affected across all the events pertaining to a specific event class.

4.3. pktcEventThrottle

As indicated earlier, the generated events can be stored locally or transmitted using SNMP, syslog, or both. However, the management stations receiving such events may wish to control the rate of transmission of such events. This event-throttling behavior is provided by the MIB objects in this sub-tree.

Some highlights are as follows:

pktcEventThrottleAdminStatus - this MIB object allows for transmissions to be unconstrained, maintained below threshold, stopped at the threshold, or inhibited.

pktcEventThrottleThreshold - this MIB object specifies the throttle, i.e., the number of events over an interval that is considered to be the threshold.

pktcEventThrottleInterval - this MIB object specifies the interval over which the threshold is calculated.

4.4. pktcEventStatus

This sub-tree is designed to provide status information related to event transmissions. It currently contains one MIB object, pktcEventTransmissionStatus, that allows a client to report the status of event transmissions.

4.5. pktcEvent

This sub-tree is designed to provide a list of all the events that can be generated by an MTA and its associated descriptions. The MIB objects are grouped under the MIB table pktcEventTable.

4.6. pktcEventLog

This sub-tree is designed to allow the MTA to store all the events that are generated during its operation. The events are stored with information such as the time of the event, its description and related characteristics like severity levels.

4.7. pktcEventNotifications

This sub-tree specifies the notification information, i.e., when MTAs transmit messages using SNMP traps and informs. SNMP traps refer to the SNMPv2-Trap-PDU. SNMPv1 traps are disallowed.

5. Relationship to Other MIB Modules

Some management objects defined in other MIB modules are applicable to an entity implementing this MIB. In particular, it is assumed that an entity implementing the PKTC-IETF-EVENT-MIB module will also implement the 'interfaces' group of the IF-MIB [RFC2863].

5.1. MIB Modules Required for IMPORTS

The PKTC-IETF-EVENT-MIB MIB module IMPORTS objects from SNMPv2-SMI [RFC2578], SNMPv2-TC [RFC2579], SNMP-FRAMEWORK-MIB [RFC3411], SNMPv2-CONF [RFC2580], IF-MIB [RFC2863], INET-ADDRESS-MIB [RFC4001], SNMP-TARGET-MIB [RFC3413], SNMP-NOTIFICATION-MIB [RFC3413], and the SYSLOG-TC-MIB [RFC5427].

6. Definitions

```

PKTC-IETF-EVENT-MIB DEFINITIONS ::= BEGIN

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

    TruthValue,
    DateAndTime, TEXTUAL-CONVENTION     FROM SNMPv2-TC
    SnmpAdminString                       FROM SNMP-FRAMEWORK-MIB
    OBJECT-GROUP,
    MODULE-COMPLIANCE,
    NOTIFICATION-GROUP                   FROM SNMPv2-CONF
    ifPhysAddress                         FROM IF-MIB
    InetAddressType,
    InetAddress,
    InetPortNumber                       FROM INET-ADDRESS-MIB
    snmpTargetBasicGroup, snmpTargetResponseGroup
                                         FROM SNMP-TARGET-MIB
    snmpNotifyGroup, snmpNotifyFilterGroup
                                         FROM SNMP-NOTIFICATION-MIB
    SyslogSeverity, SyslogFacility       FROM SYSLOG-TC-MIB;

pktcIetfEventMib MODULE-IDENTITY
    LAST-UPDATED "200903300000Z" -- 30 March 2009
    ORGANIZATION "IETF IP over Cable Data Network Working Group"
    CONTACT-INFO
        "Sumanth Channabasappa
         Cable Television Laboratories, Inc.
         858 Coal Creek Circle,
         Louisville, CO 80027, USA
         +1 303-661-3307
         Sumanth@cablelabs.com

         Wim De Ketelaere
         tComLabs
         Gildestraat 8
         9000 Gent, Belgium
         +32 9 269 22 90
         deketelaere@tComLabs.com"

```


Eugene Nechamkin
Broadcom Corporation
200 - 13711 International Place
Richmond, BC, V6V 2Z8, Canada
+1 604 233 8500
enechamkin@broadcom.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-Chair: Jean-Francois Mule, jf.mule@cablelabs.com
Co-Chair: Richard Woundy, Richard_Woundy@cable.comcast.com"

DESCRIPTION

"This MIB module specifies the basic management objects for managing events generated by the Multimedia Terminal Adapter devices compliant with the PacketCable and IPCablecom requirements.

Copyright (c) 2009 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of Internet Society, IETF or IETF Trust, nor the names of specific contributors, may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS 'AS IS' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES

(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This version of this MIB module is part of RFC 5428; see the RFC itself for full legal notices."

REVISION "200903300000Z" -- 30 March 2009

DESCRIPTION

"Initial version, published as RFC 5428."

::= { mib-2 182 }

SyslogSeverityMask ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"This textual convention represents a bit mask representing the severity of the syslog events that can be generated. It corresponds to the various severity levels associated with syslog messages, as specified in 'The Syslog Protocol', [RFC5424].

emerg	(0),	- emergency; system is unusable
alert	(1),	- action must be taken immediately
crit	(2),	- critical condition
err	(3),	- error condition
warning	(4),	- warning condition
notice	(5),	- normal but significant condition
info	(6),	- informational message
debug	(7)	- debug-level messages"

SYNTAX BITS {
 emerg(0),
 alert(1),
 crit(2),
 err(3),
 warning(4),
 notice(5),
 info(6),
 debug(7)
 }

```

--
--
pktcEventNotifications OBJECT IDENTIFIER ::= { pktcIetfEventMib 0 }
pktcEventMibObjects    OBJECT IDENTIFIER ::= { pktcIetfEventMib 1 }
pktcEventConformance  OBJECT IDENTIFIER ::= { pktcIetfEventMib 2 }
--
--
pktcEventControl       OBJECT IDENTIFIER ::= { pktcEventMibObjects 1 }
pktcEventThrottle     OBJECT IDENTIFIER ::= { pktcEventMibObjects 2 }
pktcEventStatus       OBJECT IDENTIFIER ::= { pktcEventMibObjects 3 }
pktcEvents            OBJECT IDENTIFIER ::= { pktcEventMibObjects 4 }
pktcEventLog          OBJECT IDENTIFIER ::= { pktcEventMibObjects 5 }

```

```
---
```

```
-- Event Reporting control objects
```

```
---
```

```
pktcEventReset OBJECT-TYPE
```

```

  SYNTAX BITS {
      resetEventLogTable(0),
      resetEventTable(1)
  }

```

```
MAX-ACCESS read-write
```

```
STATUS current
```

```
DESCRIPTION
```

```

    "This MIB object allows a management station to
    clear the local log of generated events, reset the
    management event descriptions, or both.

```

```

    MTAs generate management events. These events are stored
    in the MIB table pktcEventLogTable. If a management
    station needs to clear all the current entries (e.g.,
    after a troubleshooting operation is complete), it can
    do so by setting the resetEventLogTable(0) bit to a
    value of '1'.

```

```

    The MTA is pre-configured with the events that it can
    generate. This is stored in the MIB table
    pktcEventTable. This table also contains the
    descriptions associated with these events. These
    descriptions can be modified by a management station.
    However, if the management station wishes to reset the
    descriptions to factory defaults, it can do so by
    setting the resetEventTable(1) bit to a value of '1'.

```

```
The MTA actions are summarized below:
```

```

    Bit resetEventLogTable(0) set to a value of '1'
    - delete all entries in pktcEventLogTable;

```

- reset the value of pktcEventLogIndex to '0'.

Bit resetEventTable(1) set to a value of '1'

- reset the pktcEventTable to the factory default values.

Bits resetEventLogTable(0) and resetEventTable(1) set to a value of '1'

- perform the above actions as though they were performed individually (in any order).

Setting a reset bit to a value of '0' MUST NOT result in any action.

The MTA MUST perform the above actions regardless of persistence (i.e., storage in non-volatile memory).

The MTA MUST always return a value of '00' when this MIB object is read.

A management station that resets tables using this MIB object needs to be careful about the impact to other management stations that may be reliant on the information contained in the table(s) being reset. For example, say management station A creates a specific set of event descriptions in the event table (pktcEventTable) for debugging purposes and expects any generated events to report the modified descriptions. In such a case, if another management station resets the event table to factory defaults, any subsequent events will not contain the modified descriptions expected by management station A. Such multi-manager contentions are not addressed within this MIB module. Thus, management stations are RECOMMENDED to use this MIB object with care and caution, and only when absolutely required."

```
::= { pktcEventControl 1 }
```

```
---
```

```
-- syslog-specific MIB objects
```

```
---
```

```
pktcEventSyslog OBJECT IDENTIFIER ::= { pktcEventControl 2 }
```

```
pktcEventSyslogCapabilities OBJECT-TYPE
```

```
SYNTAX BITS {
    formatBSDSyslog(0),
    formatSyslogProtocol(1),
    transportUDP(2),
```

```

        transportTLS(3),
        transportBEEP(4)
    }
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
    "This MIB object contains the MTA capabilities
    for supporting the syslog protocol, specifically
    the message formats and the transport protocols.

    The BSD syslog message format is specified
    in [RFC3164] (formatBSDSyslog), and the IETF
    syslog protocol is specified in [RFC5424]
    (formatSyslogProtocol).

    The MTA MUST set the appropriate protocol and
    transport bits, based on implementation."
REFERENCE
    "The BSD syslog Protocol, [RFC3164];
    The Syslog Protocol, [RFC5424];
    Transmission of Syslog Messages over UDP, [RFC5426];
    TLS Transport Mapping for Syslog, [RFC5425];
    Reliable Delivery for syslog, [RFC3195]."
 ::= { pktcEventSyslog 1 }

```

```

pktcEventSyslogAddressType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "This MIB object defines the Internet address type of
        the syslog server specified by the MIB object
        pktcEventSyslogAddress.  A value of dns(16) is
        disallowed since a non-resolvable DNS domain name
        will leave the device without a syslog server to
        which it can report events."
    REFERENCE
        "PacketCable MTA Device Provisioning Specification,
        [PKT-SP-PROV]."
    DEFVAL { ipv4 }
 ::= { pktcEventSyslog 2 }

```

```

pktcEventSyslogAddress OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "This MIB object contains the IP address of the

```

syslog server to which the MTA can transmit a syslog message upon the generation of a management event. The type of address this object represents is defined by the MIB object pktDevEventSyslogAddressType.

The format of the syslog message is specified by the MIB object pktcEventSyslogMessageFormat."

REFERENCE

"PacketCable MTA Device Provisioning Specification, [PKT-SP-PROV];
PacketCable Management Event Mechanism Specification, [PKT-SP-MEM1.5];"

DEFVAL { "0.0.0.0" }
 ::= { pktcEventSyslog 3 }

pktcEventSyslogMessageFormat OBJECT-TYPE

SYNTAX INTEGER {
 formatBSDSyslog(1), -- The BSD syslog Protocol
 formatSyslogProtocol(2) -- The syslog Protocol
 }

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This MIB object contains the syslog message format to be used for transmitting syslog messages to the server contained in the MIB object pktcEventSyslogServer."

REFERENCE

"The BSD syslog Protocol, [RFC3164];
The Syslog Protocol, [RFC5424]."

DEFVAL { formatSyslogProtocol }
 ::= { pktcEventSyslog 4 }

pktcEventSyslogTransport OBJECT-TYPE

SYNTAX INTEGER {
 udp(1), -- Transmission of syslog messages over UDP
 tls(2), -- TLS Transport Mapping for Syslog
 beep(3) -- BEEP Transport Mapping for Syslog
 }

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This MIB object specifies the transport to be used to transmit syslog messages to the syslog server contained in the MIB object pktcEventSyslogAddress.

If the MTA does not support the transport specified in a SET operation, then the

MTA MUST return an appropriate error response, such as 'inconsistentValue'."

REFERENCE

"Transmission of Syslog messages over UDP, [RFC5426];
 TLS Transport Mapping for Syslog, [RFC5425]."

DEFVAL {tls}

::= { pktcEventSyslog 5 }

pktcEventSyslogPort OBJECT-TYPE

SYNTAX InetPortNumber

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This MIB object contains the port number of the syslog server to which the syslog messages are to be transmitted."

REFERENCE

"Transmission of Syslog Messages over UDP, [RFC5426];
 TLS Transport Mapping for Syslog, [RFC5425]."

DEFVAL { 6514 }

::= { pktcEventSyslog 6 }

-- Event classes

pktcEventClassTable OBJECT-TYPE

SYNTAX SEQUENCE OF PktcEventClassEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This MIB table allows for management events that can be generated by an MTA to be classified into categories, or 'event classes'. For example, all the configuration-related events can be associated with an event class titled 'configuration'. Such a classification allows for a management station to affect changes on a common group of events at once. Two operations are specified on an event class: enabling or disabling of all the events in an event class, and selective enabling or disabling based on the severity level."

::= { pktcEventControl 3 }

pktcEventClassEntry OBJECT-TYPE

SYNTAX PktcEventClassEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Each entry in this table specifies an event class, a grouping of events, as identified by the MTA manufacturer. Any event associated with an event class in this table MUST be specified in the pktcEventTable.

The MTA MUST create one entry (index=100) for the event class titled 'generic'. This event class MUST contain all the events that are not contained in any other vendor-specified event classes.

A management station SHOULD NOT associate an event with multiple event classes. However, if an event is associated with multiple event classes, the MTA MUST give precedence to the event class with the lowest index. Thus, at a given point in time, only one event class is applicable for an event.

The event table (pktcEventTable) provides the event class that affects the event. Whenever an event is generated, the MTA MUST verify the applicable event class entry to take any specified actions.

Entries in this table persist across resets and reboots."

```
INDEX { pktcEventClassIndex }
 ::= { pktcEventClassTable 1 }
```

```
PktcEventClassEntry ::= SEQUENCE {
    pktcEventClassIndex      Unsigned32,
    pktcEventClassName      SnmpAdminString,
    pktcEventClassStatus    TruthValue,
    pktcEventClassSeverity  SyslogSeverityMask
}
```

```
pktcEventClassIndex OBJECT-TYPE
    SYNTAX      Unsigned32 (1..100)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This MIB object is an index into the event
         class table. It is a locally meaningful
         value."
    ::= { pktcEventClassEntry 1 }
```

```
pktcEventClassName OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE (1..100))
    MAX-ACCESS  read-only
```



```
STATUS      current
DESCRIPTION
  "This MIB object contains the name of the
  event class.

  Vendors MAY define different event classes
  (e.g., DHCP, SNMP, DEBUG) to group together
  management events of a particular category.

  Event class names need to take into
  consideration the SnmpAdminString definition
  requirements, such as the use of control code
  sequence CR LF to represent a newline."
 ::= { pktcEventClassEntry 2 }
```

```
pktcEventClassStatus OBJECT-TYPE
SYNTAX      TruthValue
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
  "This MIB object indicates if events belonging
  to the corresponding event class are enabled
  or disabled, for event reporting.

  Setting this object to a value of 'true' enables
  reporting of all the events in the event class.

  When enabled, the means of reporting events is
  specified by the MIB object pktcEventReporting.

  Setting this object to a value of 'false' disables
  any event reporting, irrespective of the value of the
  MIB object pktcEventReporting for a specific
  event.

  The default value of this MIB object is vendor-
  specific. However, the vendor SHOULD enable all
  event categories defined by PacketCable or
  IPCablecom by default."
 ::= { pktcEventClassEntry 3 }
```

```
pktcEventClassSeverity OBJECT-TYPE
SYNTAX      SyslogSeverityMask
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
  "This MIB object defines the severity level
  of events belonging to a specific event class
```

that are enabled for event reporting.

This MIB object has no effect on the event reporting unless the MIB object `pktcEventClassStatus` is set to a value of 'true' (enabled), for the corresponding event class.

Setting a bit within the mask to a value of '1' implies that events corresponding to that severity level MUST be reported as defined by the corresponding value of 'pktcEventReporting' for events in the event class.

Setting a bit to a value of '0' implies that events corresponding to that level MUST NOT be reported, irrespective of the corresponding value of 'pktcEventReporting' for events in the event class.

It is recommended that the bits corresponding to `emerg(0)`, `alert(1)`, `crit(2)`, and `err(3)` be set to a value of '1' to ensure reporting of events requiring immediate attention."

REFERENCE

"The Syslog Protocol, [RFC5424]."
 ::= { pktcEventClassEntry 4 }

 --

Event throttling control

`pktcEventThrottleAdminStatus` OBJECT-TYPE

SYNTAX INTEGER {
 unconstrained(1),
 maintainBelowThreshold(2),
 stopAtThreshold(3),
 inhibited(4)
 }

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This MIB object controls the throttling of the transmitted messages upon generation of an event (SNMP/syslog). It does not affect local logging of events.

A value of `unconstrained(1)` causes event messages

to be transmitted without regard to the threshold settings.

A value of `maintainBelowThreshold(2)` causes event messages to be suppressed if the number of transmissions would otherwise exceed the threshold specified by `pktcEventThrottleThreshold` over the interval specified by `pktcEventThrottleInterval`.

A value of `stopAtThreshold(3)` causes event message transmission to cease once the threshold specified by `pktcEventThrottleThreshold` (over the interval specified by `pktcEventThrottleInterval`) is reached. Event generation is resumed when the value of this MIB object is modified by a management station or when the device resets or reboots.

A value of `inhibited(4)` causes all event message transmissions to be suppressed.

An event causing both an SNMP and a syslog message is still treated as a single event.

Refer to MIB objects `pktcEventThrottleThreshold` and `pktcEventThrottleInterval` for information on throttling."

```
DEFVAL { unconstrained }
 ::= { pktcEventThrottle 1 }
```

`pktcEventThrottleThreshold` OBJECT-TYPE

SYNTAX Unsigned32(0..1024)

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This MIB object contains the number of events per `pktcEventThrottleInterval` to be transmitted before throttling.

An event resulting in multiple actions (e.g., SNMP and syslog) is still treated as a single event."

```
DEFVAL { 2 }
 ::= { pktcEventThrottle 2 }
```

`pktcEventThrottleInterval` OBJECT-TYPE

SYNTAX Unsigned32(0..604800)

UNITS "seconds"

MAX-ACCESS read-write

```

STATUS      current
DESCRIPTION
    "This MIB object contains the interval over which
    the throttle threshold applies."
DEFVAL { 1 }
 ::= { pktcEventThrottle 3 }

```

```

---
-- Reporting of transmission status
---
```

pktcEventTransmissionStatus OBJECT-TYPE

```

SYNTAX      BITS {
    syslogThrottled(0),
    snmpThrottled(1),
    validsyslogServerAbsent(2),
    validSnmpManagerAbsent(3),
    syslogTransmitError(4),
    snmpTransmitError(5)
}

```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This MIB object reflects the status of the event transmissions using syslog, SNMP, or both.

If a bit corresponding to a state is set to a value of:

```

    '1', it indicates that the state is true
    '0', it indicates that the state is false

```

If the MTA is not configured with a syslog server or an SNMP Manager, the corresponding 'throttling' and 'transmit error' bits MUST be set to a value of '0'. For example, if an SNMP Manager is not configured on the MTA, the bit corresponding to validSnmpManagerAbsent(3) is set to a value of '1', and the values of the bits corresponding to snmpThrottled(1) and snmpTransmitError(5) are set to a value of '0'.

'Event throttling' is based on thresholds and the current setting of the MIB object pktcEventThrottleAdminStatus.

'Server/Manager' indicators are based on the availability of valid syslog server/SNMP Managers.

Transmit errors are reported when detected. If an MTA cannot detect an error situation, the value of the BIT will be set '0'.

It is to be noted that not all the conditions that are indicated by this MIB object are detectable by all devices, and when detected may not be accurate. It is meant to provide a report of the status as determined by the device during event transmissions."

```
::= { pktcEventStatus 1 }
```

```
---
```

```
-- Description of events
```

```
---
```

```
pktcEventTable OBJECT-TYPE
```

```
SYNTAX SEQUENCE OF PktcEventEntry
```

```
MAX-ACCESS not-accessible
```

```
STATUS current
```

```
DESCRIPTION
```

"This MIB table contains all possible management events that can be generated by the device. This includes PacketCable- and IPCablecom-defined events and vendor-specific events."

```
::= { pktcEvents 1 }
```

```
pktcEventEntry OBJECT-TYPE
```

```
SYNTAX PktcEventEntry
```

```
MAX-ACCESS not-accessible
```

```
STATUS current
```

```
DESCRIPTION
```

"An entry in this table is created for each event the MTA implementing this MIB is capable of reporting. Entries in this table are persisted across resets and reboots."

```
INDEX { pktcEventOrganization, pktcEventIdentifier }
```

```
::= { pktcEventTable 1 }
```

```
PktcEventEntry ::= SEQUENCE {
```

```
  pktcEventOrganization Unsigned32,
```

```
  pktcEventIdentifier Unsigned32,
```

```
  pktcEventFacility SyslogFacility,
```

```
  pktcEventSeverityLevel SyslogSeverity,
```

```
  pktcEventReporting BITS,
```

```
  pktcEventText SnmpAdminString,
```

```
  pktcEventClass SnmpAdminString
```

```
}
```

```
pktcEventOrganization OBJECT-TYPE
    SYNTAX      Unsigned32(1..4294967295)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This MIB object provides the IANA enterprise number of
        the organization defining the event.  Thus, all
        PacketCable- or IPCablecom-defined events will contain
        the PacketCable or IPCablecom IANA enterprise
        number, and all vendor-specific events will contain
        the IANA enterprise number of the defining
        organization."
    REFERENCE
        "IANA Private Enterprise Number assignment,
        [IANA-ENTERPRISE]."
```

```
 ::= { pktcEventEntry 1 }
```



```
pktcEventIdentifier OBJECT-TYPE
    SYNTAX      Unsigned32(1..4294967295)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This MIB object contains the event identifier for the
        corresponding event."
    REFERENCE
        "PacketCable Management Event Mechanism Specification,
        [PKT-SP-MEM1.5];
        PacketCable MTA Device Provisioning Specification,
        [PKT-SP-PROV]."
```

```
 ::= { pktcEventEntry 2 }
```



```
pktcEventFacility OBJECT-TYPE
    SYNTAX      SyslogFacility
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This MIB object contains the facility
        for the event.
        For PacketCable, IPCablecom, or ETSI events,
        this MUST be set to a value of local0(16)."
```

```
 REFERENCE
        "The Syslog Protocol, [RFC5424];
        Textual Conventions for Syslog Management,
        [RFC5427]."
```

```
 ::= { pktcEventEntry 3 }
```



```
pktcEventSeverityLevel OBJECT-TYPE
    SYNTAX      SyslogSeverity
```

```

MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    "This MIB object contains the severity level that
    is applicable to the specified event."
REFERENCE
    "The Syslog Protocol, [RFC5424];
    Textual Conventions for Syslog Management,
    [RFC5427]."
```

::= { pktcEventEntry 4 }

pktcEventReporting OBJECT-TYPE

```

SYNTAX      BITS {
    local(0),
    syslog(1),
    snmpTrap(2),
    snmpInform(3)
}
```

```

MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
```

"This MIB object defines the action to be taken on occurrence of this event. Bit local(0) refers to local logging of events; bit syslog(1) refers to the transmission of events using syslog; bit snmpTrap(2) refers to the transmission of events using SNMP Traps (SNMPv2-Trap-PDU); and bit snmpInform(3) refers to the transmission of events using SNMP INFORMs.

Setting a bit to a value of '1' indicates that the corresponding action will be taken upon occurrence of this event. If none of the bits are set, then no action is taken upon occurrence of the event. The success of transmission using syslog and SNMP depends on the MTA configuration. For example, a valid syslog server address is required for syslog message transmission.

Specification of a management event does not necessarily include the actions to be taken upon its generation, i.e., it does not need to specify if a generated event needs to be transmitted via SNMP or syslog, or stored locally. Thus, certain default values are specified, based on the event's severity level specified by the MIB object pktcEventSeverityLevel, as follows:

- If the severity level of an event is emerg(0), alert(1), crit(2), or err(3), set the bits for local(0), syslog(1), and snmpInform(3) to a value of '1' and set the remaining bits to a value of '0'.

```

        - For an event with any other severity level, set
        the bits for local(0) and syslog(1) to a value
        of '1' and set the rest of the bits to a value
        of '0'."
 ::= { pktcEventEntry 5 }

pktcEventText OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE (0..127))
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "This MIB object provides a human-readable
        description of the event.  Descriptions need
        to take into consideration the SnmpAdminString
        definition requirements such as the use of
        control code sequence CR LF to represent a
        newline."
 ::= { pktcEventEntry 6 }

pktcEventClass OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE (0..100))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This MIB object represents the event class
        that affects the event.  If an event is associated
        with only one event class, then its name
        (pktcEventClassName) is reported.  If an event
        is associated with more than one event class,
        then the name of the event class with the
        lowest index in the event class table
        (pktcEventClassTable) is reported.

        See the MIB table pktcEventClassTable
        for a description of event classes and usage.

        Descriptions need to take into consideration the
        SnmpAdminString definition requirements, such as
        the use of control code sequence CR LF to
        represent a newline."
 ::= { pktcEventEntry 7 }

---
-- Log of generated events
---

pktcEventLogTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF PktcEventLogEntry

```



```

MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    "This MIB table contains a log of the events
    generated by the MTA.

    A description of all the events that can be
    generated by the device can be obtained from the
    MIB table pktcEventTable.

    An MTA is not required to persist the contents of this
    table across resets."
 ::= { pktcEventLog 1 }

```

```

pktcEventLogEntry OBJECT-TYPE
    SYNTAX PktcEventLogEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Each entry in this table describes an event that
        has occurred, indexed in the chronological order of
        generation. The details of the event are borrowed
        from the parameters associated with the corresponding
        event entry in pktcEventTable at the
        time of the event generation.
        While all entries created as such can be cleared using
        the MIB object pktcEventReset, the event entries
        themselves cannot be individually deleted."

    INDEX { pktcEventLogIndex }
 ::= { pktcEventLogTable 1 }

```

```

PktcEventLogEntry ::= SEQUENCE {
    pktcEventLogIndex      Unsigned32,
    pktcEventLogTime       DateAndTime,
    pktcEventLogOrganization Unsigned32,
    pktcEventLogIdentifier Unsigned32,
    pktcEventLogText       SnmpAdminString,
    pktcEventLogEndpointName SnmpAdminString,
    pktcEventLogType       BITS,
    pktcEventLogTargetInfo SnmpAdminString,
    pktcEventLogCorrelationId Unsigned32,
    pktcEventLogAdditionalInfo SnmpAdminString
}

```

```

pktcEventLogIndex OBJECT-TYPE
    SYNTAX Unsigned32 (1..4294967295)
    MAX-ACCESS not-accessible

```

STATUS current

DESCRIPTION

"This MIB object provides relative ordering of the objects in the event log.

If the MTA implements non-volatile storage, then this object will always increase except when the MIB object reaches a value of $2^{32}-1$.

If the MTA does not implement non-volatile storage, then this object will always increase except when the MIB object reaches a value of $2^{32}-1$ or the MTA is reset.

When the value reaches $2^{32}-1$, or an MTA that does not implement non-volatile storage is reset, newer events will be stored starting with an index value of '1' (cyclic rotation)."

::= { pktcEventLogEntry 1 }

pktcEventLogTime OBJECT-TYPE

SYNTAX DateAndTime

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This MIB object provides a human-readable description of the date and time at which the event occurred. The value of the date and time contained in this MIB object SHOULD reflect the date and time used in the syslog message resulting from the associated event, if such a syslog message was transmitted."

::= { pktcEventLogEntry 2 }

pktcEventLogOrganization OBJECT-TYPE

SYNTAX Unsigned32(1..4294967295)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This MIB object provides the IANA enterprise number of the organization defining the event. Thus, all PacketCable- or IPCablecom-defined events will contain the CableLabs or IPCablecom IANA enterprise number, and all vendor-specific events will contain the IANA enterprise number of the defining organization."

::= { pktcEventLogEntry 3 }

pktcEventLogIdentifier OBJECT-TYPE

SYNTAX Unsigned32


```

        local(0),
        syslog(1),
        snmpTrap(2),
        snmpInform(3)
    }
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION   "This MIB object contains the type of actions taken by
              the MTA when the event indicated by the MIB object
              pktcEventLogIdentifier occurred.

              A bit with a value of '1' indicates the corresponding
              action was taken. Setting it to a value of '0'
              indicates that the corresponding action was not taken.

              An event may trigger one or more actions (e.g., syslog
              and SNMP) or result only in a local log. An action may
              also be prevented due to throttling, in which case it is
              not reported by this MIB object."
 ::= { pktcEventLogEntry 7 }

```

pktcEventLogTargetInfo OBJECT-TYPE

```

SYNTAX        SnmpAdminString (SIZE (0..255))
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION   "This MIB object contains a comma-separated list of the
              actions taken for external notifications, along with the
              target IP address for the generated events. Locally
              stored events MUST NOT be recorded in this MIB object.

```

The syntax is as:

```
<action-1/IP>,<action-2/IP>,<action-3/IP>
```

Where <action-n/IP> is to be denoted as follows:

For syslog events:

```
syslog/<IP address of the syslog server>
```

For SNMP traps:

```
snmpTrap/<IP address of the SNMP server>
```

For SNMP INFORMS:

```
snmpInform/<IP address of the SNMP server>
```

If there are multiple targets for the same type (SNMP traps sent to multiple IP addresses) or if there are multiple message types sent to the same IP (syslog and SNMP sent to the same IP address), they need to be reported individually.

It is to be noted that this MIB object may not be able to store all the data in some cases (e.g., multiple IPv6 addresses), in which case some actions may not be reported. In such cases, the MTA MUST present a value of '.' at the end of the value.

Values contained by this MIB object need to observe the SnmpAdminString definition requirements."

```
::= { pktcEventLogEntry 8 }
```

pktcEventLogCorrelationId OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This MIB object contains the correlation ID generated by the MTA during the initiation of the last provisioning flow, within or following which the event occurred.

Although a correlation ID once generated after MTA reset does not change until next MTA reset, the value of this object will differ for the events preserved across MTA resets in case of a persistent pktcEventLogTable.

For more information on the generation of correlation IDs, refer to the corresponding PacketCable/IPCablecom Device Provisioning specifications."

REFERENCE

"PacketCable MTA Device Provisioning Specification, [PKT-SP-PROV]."

```
::= { pktcEventLogEntry 9 }
```

pktcEventLogAdditionalInfo OBJECT-TYPE

SYNTAX SnmpAdminString (SIZE (0..255))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This MIB object contains additional information in relation to the corresponding event that an MTA might wish to report, such as parameterized data or debugging information. The format is vendor-specific.

If the MTA cannot provide any additional information for the particular event generated, it MUST populate this

```

MIB object with a zero-length OCTET-STRING. Vendors
providing this information need to observe the
SnmpAdminString definition requirements, such as the
use of control code sequence CR LF for newline."
 ::= { pktcEventLogEntry 10 }

---
-- Notifications
---

pktcEventNotification NOTIFICATION-TYPE
  OBJECTS {
    pktcEventLogTime,
    pktcEventLogOrganization,
    pktcEventLogIdentifier,
    pktcEventLogEndpointName,
    pktcEventLogCorrelationId,
    ifPhysAddress
  }
  STATUS      current
  DESCRIPTION
    "This Notification MIB object contains the contents for
    event reporting.

    It contains the event log time, the organization
    ID, the event identifier, the endpoint identifier, the
    correlation ID, and the MTA's MAC address."
  ::= { pktcEventNotifications 1 }

---
-- Conformance/Compliance
---

pktcEventCompliances OBJECT IDENTIFIER ::=
  { pktcEventConformance 1 }
pktcEventGroups      OBJECT IDENTIFIER ::=
  { pktcEventConformance 2 }

pktcEventBasicCompliance MODULE-COMPLIANCE
  STATUS      current
  DESCRIPTION
    "The compliance statement for devices that implement
    the event-reporting feature."

  MODULE     --pktcIetfEventMib

  MANDATORY-GROUPS {
    pktcEventGroup,

```

```
        pktcEventNotificationGroup
    }

MODULE SNMP-TARGET-MIB
    MANDATORY-GROUPS {
        snmpTargetBasicGroup,
        snmpTargetResponseGroup
    }

MODULE SNMP-NOTIFICATION-MIB
    MANDATORY-GROUPS {
        snmpNotifyGroup,
        snmpNotifyFilterGroup
    }

 ::= { pktcEventCompliances 3 }

pktcEventGroup OBJECT-GROUP
    OBJECTS {
        pktcEventReset,
        pktcEventSyslogCapabilities,
        pktcEventSyslogAddressType,
        pktcEventSyslogAddress,
        pktcEventSyslogTransport,
        pktcEventSyslogPort,
        pktcEventSyslogMessageFormat,
        pktcEventThrottleAdminStatus,
        pktcEventThrottleThreshold,
        pktcEventThrottleInterval,
        pktcEventTransmissionStatus,
        pktcEventFacility,
        pktcEventSeverityLevel,
        pktcEventReporting,
        pktcEventText,
        pktcEventLogTime,
        pktcEventLogOrganization,
        pktcEventLogIdentifier,
        pktcEventLogText,
        pktcEventLogEndpointName,
        pktcEventLogType,
        pktcEventLogTargetInfo,
        pktcEventLogCorrelationId,
        pktcEventLogAdditionalInfo,
        pktcEventClass,
        pktcEventClassName,
        pktcEventClassStatus,
        pktcEventClassSeverity
    }
}
```

```

STATUS      current
DESCRIPTION
    "Group of MIB objects for PacketCable Management Event
    MIB."
 ::= { pktcEventGroups 1 }

pktcEventNotificationGroup NOTIFICATION-GROUP
NOTIFICATIONS { pktcEventNotification }
STATUS      current
DESCRIPTION
    "Group of MIB objects for notifications related to
    change in status of the MTA Device."
 ::= { pktcEventGroups 2 }
END

```

7. IANA Considerations

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

Descriptor	OBJECT IDENTIFIER Value
-----	-----
pktcIetfEventMib	{ mib-2 182 }

8. Security Considerations

There are a number of management objects defined in this MIB module with a MAX-ACCESS clause of read-write. 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. Security threats include events unreported on errors, redirection of events (deliberately or otherwise) or minimized reporting of errors. Such threats can mask certain misconfiguration attempts and denial of service attacks that can be recognized and thwarted via event reporting.

MIB objects of significance include:

- those that control the event generation, the target syslog address for events and the reporting status, i.e.:
 - pktcEventReset
 - pktcEventSyslogAddressType
 - pktcEventSyslogAddress
 - pktcEventSyslogPort
 - pktcEventSyslogMessageFormat
 - pktcEventSyslogTransport
 - pktcEventClassStatus
- those related to event classes, i.e.: pktcEventClassSeverity
- those related to throttling, i.e.: pktcEventThrottleAdminStatus
pktcEventThrottleThreshold pktcEventThrottleInterval
- those related to the event reporting capabilities of an MTA, i.e.:
pktcEventSeverityLevel pktcEventReporting pktcEventText

The MIB object `pktcEventReset` deserves special mention since access to this MIB object can be used to disrupt event collection by management stations. For example, consider a management station that modifies the descriptions in the event table `pktcEventTable`. It would then expect management events generated by the MTA to reflect the modified values. A rogue management station that has access to the `pktcEventReset` can reset the event table, resulting in the management station not receiving events with the expected descriptions. Further, a rogue management station with access to `pktcEventReset` can also clear local logs, eliminating local logs of generated events for management stations that are not configured to receive syslog or SNMP messages. The same concerns apply when allowed management stations performing such operations are unaware of other management stations that may be reliant on the event table or the event log table for management or monitoring. This MIB module does not address such multi-manager contentions, and recommends that the MIB object `pktcEventReset` be used with caution.

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

pktcEventLogTable: This table contains the log of generated event messages. Read access to this table might reveal some specific information that should be kept confidential.

pktcEventTransmissionStatus: This MIB object reveals the status of event transmission and MAY be sensitive in some environments.

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

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

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

9. Acknowledgments

The authors would like to thank the members of the IETF IP over Cable Data Network (IPCDN) working group and the CableLabs PacketCable Provisioning focus team for their contributions, comments, and suggestions.

Special appreciation is extended to the following individuals (in alphabetical order): Dan Romascanu, David Harrington, Greg Nakanishi, Jean-Francois Mule, John Berg, Kevin Marez, Paul Duffy, Peter Bates, Randy Presuhn, Rich Woundy, Rick Vetter, Roy Spitzer, and Satish Kumar.

The primary editor (Sumanth) wishes to acknowledge the MIB doctors David Harrington and Dan Romascanu, Lars Eggert and Pasi Eronen, as well as Rich Woundy for expert feedback and numerous suggestions to improve this document.

10. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [PKT-SP-PROV] Packetcable MTA Device Provisioning Specification, PKT-SP-PROV-I11-050812.
- [RFC3413] Levi, D., Meyer, P., and B. Stewart, "Simple Network Management Protocol (SNMP) Applications", STD 62, RFC 3413, December 2002.
- [RFC5424] Gerhards, R., "The Syslog Protocol", RFC 5424, March 2009.
- [RFC5426] Okmianski, A., "Transmission of Syslog Messages over UDP", RFC 5426, March 2009.
- [RFC5425] Miao, F., Ed., Ma, Y., Ed., and J. Salowey, Ed., "Transport Layer Security (TLS) Transport Mapping for Syslog", RFC 5425, March 2009.
- [RFC5427] Keeni, G., "Textual Conventions for Syslog Management", RFC 5427, March 2009.
- [RFC3195] New, D. and M. Rose, "Reliable Delivery for syslog", RFC 3195, November 2001.
- [ITU-T-J176] IPCablecom Management Event Mechanism MIB, J.176, ITU-T, August 2002.
- [PKT-SP-EVEMIB1.5] PacketCable(TM) Management Event MIB Specification, PKT-SP-EVEMIB1.5-I02-050812, August, 2005.
- [PKT-SP-MEM1.5] PacketCable(TM) Management Event Mechanism Specification, PKT-SP-MEM1.5-I02-050812, August, 2005.
- [ETSITS101909-22] ETSI TS 101 909-22, "Digital Broadband Cable Access to the Public Telecommunications Network", IP Multimedia Time Critical Services, Part 22, Management Event Messages.
- [RFC768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

- [RFC2578] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.
- [RFC2579] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.
- [RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.
- [RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.
- [RFC3411] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", STD 62, RFC 3411, December 2002.
- [RFC4001] Daniele, M., Haberman, B., Routhier, S., and J. Schoenwaelder, "Textual Conventions for Internet Network Addresses", RFC 4001, February 2005.
- [IANA-ENTERPRISE] "IANA Private Enterprise Numbers", <http://www.iana.org/>

11. Informative References

- [RFC3164] Lonvick, C., "The BSD Syslog Protocol", RFC 3164, August 2001.
- [RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.
- [PKT-SP-MGCP] Packetcable Network-Based Call Signaling Protocol Specification, PKT-SP-EC-MGCP-I11-050812.
- [RFC3435] Andreasen, F. and B. Foster, "Media Gateway Control Protocol (MGCP) Version 1.0", RFC 3435, January 2003.
- [RFC4682] Nechamkin, E. and J-F. Mule, "Multimedia Terminal Adapter (MTA) Management Information Base for PacketCable- and IPCablecom-Compliant Devices", RFC 4682, December 2006.

[PKT-SP-CODEC] Packetcable Audio/Video Codecs Specification,
PKT-SP-CODEC-I06-050812.

Authors' Addresses

Sumanth Channabasappa
Cable Television Laboratories, Inc.
858 Coal Creek Circle,
Louisville, CO 80027, USA

Phone: +1 303-661-3307
EMail: Sumanth@cablelabs.com

Wim De Ketelaere
tComLabs
Gildestraat 8
9000 Gent, Belgium

Phone: +32 9 269 22 90
EMail: deketelaere@tComLabs.com

Eugene Nechamkin
Broadcom Corporation
200 - 13711 International Place
Richmond, BC, V6V 2Z8, Canada

Phone: +1 604 233 8500
EMail: enechamkin@broadcom.com

