1. INTRODUCTION

This document outlines the Radio Frequency Interface (RFI) Management Information Bases (MIBs) for high-speed data-over-cable systems developed by the DOCSIS Data Over Cable Services working group.

Two Simple Network Management Protocol (SNMP) MIBs are defined. The first is the DOCSIS Radio Frequency Interface (RFI) MIB and defines objects that enable management of the CATV MAC and PHY layer interfaces. The second is the DOCSIS Cable Device (CD) MIB and defines objects that enable management of CMs and Cable Modem Termination Systems (CMTSs).

This specification is intended to enable prospective vendors of cable modems and other data-over-cable systems to address the operations support requirements in a uniform and consistent manner.

1.1. Requirements

Throughout this document, the words that are used to define the significance of particular requirements are capitalized. These words are:

“MUST”	This word or the adjective “REQUIRED” means that the item is an absolute requirement of this specification.
“MUST NOT”	This phrase means that the item is an absolute prohibition of this specification.
“SHOULD”	This word or the adjective “RECOMMENDED” means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully

	weighed before choosing a different course.
“SHOULD NOT”	This phrase means that there may exist valid reasons in particular circumstances when the listed behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
“MAY”	This word or the adjective “OPTIONAL” means that this item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because it enhances the product, for example; another vendor may omit the same item.

2. CM AND CMTS MANAGEMENT REQUIREMENTS

This section describes the CM and CMTS management requirements. The MIBs compliant with these requirements are described in Section 4 and formally defined in Section 5.

2.1. Accounting Management

Although many different types of billing scenarios exist for operators, the only scenarios which require use of CM and CMTS managed objects are those based on metered usage or reserved bandwidth. Common practice by several Internet Service Providers (ISPs) allows usage-based billing based on peak rates. A DOCSIS provider can implement usage-based billing two ways: by polling the CMs, or by polling the CMTS.

In the first method, a service provider can poll the ifInOctets and ifOutOctets counters from the MIB-II [RFC-1213] Interfaces group on each CM. This has the advantage of enabling both upstream and downstream traffic metering with the potential disadvantage of affecting network performance.

The second metered billing method involves monitoring the docsIfCmtsServiceTotalDataSlots counter from the docsIfCmtsServiceTable on each CMTS. This has the advantage of avoiding congestion on the RF network; however, it enables upstream traffic metering only. In a typical ISP environment, a BSS polls the appropriate counters on each customer device once every 15 minutes throughout a monthly billing cycle. This data is converted into an average utilization rate for the sample period. Doing so permits the ISP to bill based on peak bandwidth by choosing the sample ranked at the 90-95th percentile. Note that the billing system may also include time-of-day rate variations. The billing of reserved upstream MAC bandwidth is aided by information available from the docsIfQosProfileTable for each CM. These MIB variables report the upstream QoS characteristics, not just the nominal bandwidth, associated with each service ID and enable the service provider to bill for Grade of Service by verifying QoS.

2.2. Configuration Management

2.2.1. Version Control

The CM and CMTS SHOULD support software revision and operational parameter configuration interrogation. In particular, the fields of the sysObjectID Object Identifier (OID) of the CM should successively encode the vendor ID, the hardware platform, the hardware revision, the software/PROM major revision number, the software/PROM minor revision number, and (optionally) the software patch level. Each parameter MUST occupy exactly one field. The fields of the sysObjectID OID of the CMTS SHOULD use the same encoding.

Additionally, the CM MUST (and the CMTS SHOULD) include the same revision information in the vendor defined text of the sysDescr object in the MIB-II System Group [RFC-1213].

2.2.2. Software upgrades

The CM software upgrade process is documented in [DOCSIS7].

The mechanism to upgrade software from an SNMP manager MUST be supported by CMs, and SHOULD be supported by CMTSs. From a network management station, the operator:

- sets docsDevSwServer to the address of the TFTP server for software upgrades
- sets docsDevSwFilename to the file pathname of the software upgrade image
- sets docsDevSwAdminStatus to upgrade-from-mgt

docsDevSwAdminStatus MUST persist across resets/reboots until overwritten from an SNMP manager or via the CM configuration file.

The default state of docsDevSwAdminStatus MUST be allowProvisioningUpgrade(2) until it is overwritten by ignoreProvisioningUpgrade(3) following a successful SNMP-initiated software upgrade.

docsDevSwOperStatus MUST persist across resets to report the outcome of the last software upgrade attempt.

Both docsDevSwServer and docsDevSwFilename MUST behave according to their textual descriptions in the cable device MIB.

If a CM suffers loss of power or resets during an SNMP-initiated upgrade, the CM MUST resume the interrupted upgrade without requiring manual intervention.

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 modem without changing the provisioning database.

Note that software upgrades MUST NOT be accepted blindly by the cable modem. The processes defined in [DOCSIS7] for CM response following software upgrade failure MUST be supported.[1]

2.2.3. System Initialization and Configuration

Most system configuration of CMs is performed through a combination of CATV MAC, DHCP, and TFTP exchanges. These exchanges are defined in detail in the Radio Frequency Interfaces Specification [DOCSIS7]. In particular, to enable event logging through SYSLOG, the DHCP server sets the log server option [RFC-2132] to the address of the SYSLOG server.

2.3. Fault Management

2.3.1. SNMP Usage

In the DOCSIS environment, the goals of fault management are the remote detection, diagnosis, and correction of network problems. Therefore, the CM MUST support SNMP management traffic across both the Ethernet and CATV MAC interfaces. Access may be restricted to support policy goals (see the docsDevNmAccessTable).

CM installation personnel can use SNMP queries from a station on the Ethernet to perform on-site CM and CATV MAC diagnostics and fault classification (note that this may require temporary provisioning of the CM from an Ethernet DHCP server).

Further, future customer applications using SNMP queries can diagnose simple post-installation problems, avoiding visits from service personnel and minimizing help desk telephone queries.

Standard MIB-II support MUST be implemented to instrument interface status, packet corruption, protocol errors, etc. The transmission MIB for Ethernet-like objects [RFC-1643] MUST be implemented on each CM and CMTS Ethernet and Fast Ethernet port. The ifXTable [RFC-2233] SHOULD be implemented to provide [1]discrimination between broadcast and multicast traffic.

The CM and CMTS MUST support managed objects for fault management of the PHY and MAC layers.

The MIB includes variables to track PHY state such as codeword collisions and corruption, signal-to-noise ratios, transmit and receive power levels, propagation delays, micro-reflections, in channel response, and Sync loss. The MIB also includes variables to track MAC state such as collisions and excessive retries for requests, immediate data transmits, and initial ranging requests.

For fault management at all layers, the CM/CMTS MUST generate replies to SNMP queries (subject to policy filters) for counters and status, MUST send SNMP traps to one or more trap NMSs (subject to policy), and MUST send event logging to a SYSLOG server (if a log server is defined). The ifTestTable [RFC-2233] SHOULD be implemented for any diagnostic test procedures that can be remotely initiated.

2.3.2. Event Logging

Event logging and history provide vendors an opportunity for product differentiation. The ability to report useful logs may depend on semi-graceful failure modes and on the ability to record such in nonvolatile storage.

[1] ECN OSS-N-99031: Clasification on software download process

Events SHOULD be reported via log entries in a MIB, the SYSLOG facility (as documented in Appendix B), and SNMP traps. Reporting of events SHOULD be fully configurable by priority class. At minimum, it MUST be possible to disable SNMP Trap and SYSLOG transmission.
A local event log that is available via SNMP queries SHOULD be implemented to track events that cannot be reported at the time that they occur. This log SHOULD support a minimum of ten event log entries, and SHOULD persist across device reboots.

The definition and coding of events is vendor-specific. However, the standard set of error codes and messages listed in Appendix I of [DOCSIS 7] SHOULD be used to textually describe events where applicable.

In deference to the network operator who must troubleshoot multi-vendor networks, the circumstances and meaning of each event are reported as human-readable text. Vendors SHOULD provide time-of-day clocks in CMs to provide useful timestamping of events. Similarly, event logs SHOULD be persistent across device reboots. The depth of the event log is vendor-dependent, with oldest entries discarded as needed.

For each vendor-specific event that is reportable via TRAP, the vendor must create an enterprise-specific trap definition. Trap definitions MUST include docsDevEvText and SHOULD be defined according to section 3.2.2. of draft-ietf-ipcdn-cable-device-mib.txt.

The event framework described in this section MUST be implemented in CMs and SHOULD be implemented in CMTSs.

2.3.3. Trap and Syslog Throttling

The CM and CMTS MUST provide support for trap and syslog message throttling as described below.

The network operator can employ message rate throttling or trap limiting by manipulating the appropriate MIB variables.

2.3.3.1. Rate Throttling

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

In the second method, the device resumes sending traps 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 trap and SYSLOG messages (excluding duplicates) to be transmitted within the measurement interval. The operator can query the operational throttling state (to determine whether traps 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.

2.3.3.2. Trap Limiting

Network operators may wish to limit the number of traps sent by a device over a specified time period.

The device ceases to send traps and SYSLOG messages when the number of traps exceeds the specified threshold. It resumes sending traps only when the measurement interval has passed.

The network operator defines the maximum number of traps 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 threshold which is the maximum number of traps.

See “Techniques for Managing Asynchronously Generated Alerts” [RFC-1224] for further information.

2.3.4. Non-SNMP Fault Management protocols

The OSS can use a variety of tools and techniques to examine faults at multiple layers. For the IP layer, useful non-SNMP based tools include ping (ICMP Echo and Echo Reply), traceroute (UDP and various ICMP Destination Unreachable flavors). Pings to a CM from its Ethernet side MUST be supported to enable local connectivity testing from a customer's PC to the modem. The CM and CMTS MUST support IP end-station generation of ICMP error messages and processing of all ICMP messages.

2.4. Performance Management

At the CATV MAC layer, performance management focuses on the monitoring of the effectiveness of cable plant segmentation and rates of upstream traffic and collisions. Instrumentation is provided in the form of the standard interfaces statistics, as well as the mcnsifCmtsServiceTable and mcnsifCmServiceTable.

It is not anticipated that the CMTS upstream bandwidth allocation function will require active network management intervention and tuning. Nevertheless, management objects are provided in case tuning or direct control is necessary. The three key upstream contention intervals are the request interval, the immediate data interval, and the initial ranging maintenance interval. If the upstream collision rate of requests and immediate data is high relative to the upstream traffic bandwidth, then the network management system (NMS) might increase the size of the request and immediate data intervals, respectively. The NMS might increase the size of the initial ranging maintenance interval when the upstream collision rate of initial ranging messages is relatively high, such as at the conclusion of a widespread regional power outage. The NMS might also decrease the size of these contention intervals under low collision rate conditions, since these intervals occupy bandwidth that may be otherwise used for upstream transmission bandwidth. As a last resort, the NMS might change the guaranteed upstream bandwidth for one or more service IDs, to relieve upstream traffic congestion for key subscribers. The CM MUST implement

MIB counters that report the number of contention interval collisions (measured as the number of contention interval retries) per service ID, and the CMTS MUST implement read-write MIB objects that control the size of the contention intervals for each upstream channel. The CMTS SHOULD implement a read-write MIB object that controls the guaranteed upstream bandwidth for each service ID.

At the LLC layer, the performance management focus is on bridge traffic management. The CM and CMTS (if the CMTS implements transparent bridging) MUST implement the Bridge MIB [RFC-1493], including the dot1dBase and dot1dTp groups. The CM and CMTS MUST implement a managed object that controls whether the 802.1d spanning tree protocol (STP) is run and topology update messages are generated; STP is unnecessary in hierarchical, loop-free topologies. If the STP is enabled for the CM/CMTS, then the CM/CMTS MUST implement the dot1dStp group. These MIB groups' objects allow the NMS to detect when bridge forwarding tables are full, and enable the NMS to modify aging timers.

A final performance concern is the ability to diagnose unidirectional loss. Both the CM and CMTS MUST implement the MIB-II [RFC-1213] Interfaces group. When there exists more than one upstream or downstream channel, the CM/CMTS MUST implement an instance of IfEntry for each channel. The ifStack MIB [RFC-2233] MUST be used to define the relationships among the CATV MAC interfaces and their channels.

2.5. Protocol Filters

The CM MUST implement 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, IPX, NetBIOS, and Appletalk). The IP protocol filter entries can be used to restrict upstream or downstream traffic based on source and destination IP addresses, transport-layer protocols (such as TCP, UDP, and ICMP), and source and destination TCP/UDP port numbers. The CM MUST support a minimum of ten LLC protocol filter entries, and ten IP protocol filter entries.

2.6. Common Spectrum Management

The CMTS SHOULD implement the HFC RF Spectrum Management MIB [CSMI-MIB]. The definition of this MIB is likely to evolve, and vendors should anticipate changes in this area.

3. AREAS FOR FUTURE CONSIDERATION

This section outlines some areas for future consideration within this specification.

- Enterprise-specific traps will be defined in the future as dictated by field experience.
- Multicast service provisioning within the cable modem will be clearly defined.
- To support the billing of reserved downstream MAC bandwidth, the CMTS should implement the evolving RSVP/Integrated Services MIB(s). Because of the variety of output queuing mechanisms, comments are solicited for the management mechanisms to support this.

4. MANAGEMENT INFORMATION BASE (MIB)

This section defines the minimum set of managed objects required to support the CM and CMTS management requirements identified in the previous section. Vendors may augment this MIB with objects from other standard or vendor-specific MIBs where appropriate.

The DOCSIS OSSI specification has priority over IETF MIB specifications. Vendors MUST implement MIB requirements in accordance with the texts specified in the OSSI specification. See Appendix C for detailed MIB requirements. Certain objects are deprecated but may be required by the OSSI specification.

Otherwise, implementation of deprecated objects MUST follow the following rules:

1. a deprecated "scalar" object MUST return 'NoSuchName'
2. a deprecated "element that is part of table" MUST return '0'

Obsolete object(s) MUST be implemented if specified by the OSSI specification.

4.1. MIB Organization

There are three parts of the MIB needed for CMs and CMTSs. The first is a set of objects drawn from generic[2] SNMP MIBs that bear on this set[3] of devices. It is not the intention of this specification to duplicate existing specifications. These are available as RFCs from the IETF and are widely available.

The second part is a set of objects for the CATV interfaces of the CM and CMTS. This MIB provides the objects needed to configure, operate, and monitor the physical CATV interfaces. This specification is derived from the DOCSIS Radio Frequency Specification [DOCSIS7]. These objects are defined in the Radio Frequency (RF) Interface MIB for DOCSIS RF interfaces [RFC-2670] and MUST be implemented.[4]

The docsIfDownChannelPower object-type MUST be implemented in a CMTS that provides an integrated RF upconverter. If the CMTS relies on an external upconverter, then the CMTS SHOULD implement the docsIfDownChannelPower object-type. The CMTS transmit power reported in the MIB object MUST be within 2 dB of the actual transmit power in dBmV when implemented. If transmit power management is not implemented, the MIB object will be read-only and report the value of 0 (zero).

The docsIfDownChannelPower object-type MUST be implemented in DOCSIS 1.0 conforming CMs. This object is read-only. When operated at nominal line voltage, at normal room temperature, the reported power MUST be within 3 dB of the actual received channel power. Across the input power range from -15 dBmV to +15 dBmV,

[2] ECN OSS-N-99082: "standard" changed to "generic"
[3] ECN OSS-N-99082: "class" changed to "set"
[4] ECN OSS-N-99082, OSS-N-99090: Refers to RFC 2670 for the DOCSIS RF MIB.

for any 1 dB change in input power, the CM MUST report a power change in the same direction that is not less than 0.5 dB and not more than 1.5 dB.

The third part is a set of objects for management of CM and CMTS devices. These provide system-level functionality that is specific to the business and operational environments of cable data systems. These objects are defined in DOCSIS Cable Device MIB [RFC-2669] and MUST be implemented.[5]

docsDevFilterLLCUnmatchedAction:

docsDevFilterLLCUnmatchedAction MUST follow the following requirement:
If set to discard(1), any L2 packet that does not match any filters will be discarded, otherwise accepted. If set to accept, any L2 packet that does not match any filters will be accepted, otherwise discarded. Another way to interpret this is the following:
action = UnMatchedAction
Iterate through the table
if there is a match (packet.protocol = row.protocol)
{ reverse the action (accept becomes discard, discard becomes accept)
apply action to the packet
terminate the iteration
}

docsDevCpeIpMax:

DOCSIS-compliant CMs MUST implement the docsDevCpeIpMax object with a default value of -1.

[5] ECN OSS-N-99082, OSS-N-99090: Refers to RFC 2669 for the DOCSIS CD MIB.

docsDevNMAccessIP and docsDevNMAccessIpMask :

The devices that implement docsDevNMAccessTable MUST apply the following rule in order to determine whether to permit SNMP access from a SrcIpAddr:

if ((NmAccessIp AND NmAccessIpMask) = = (SrcIpAddr AND NmAccessIpMask))

  Permit the access from SrcIpAddr

else

  Do NOT permit the access from SrcIpAddr

4.2. Managed Objects from Existing Standards

4.2.1. Requirements for RFC-1907

4.2.1.1. The System Group

The System Group from RFC-1907 MUST be implemented. See Section 2.2.1 for sysObjectID requirements.

4.2.1.2. The SNMP Group

The SNMP Group from RFC-1907 MUST be implemented. Obsolete objects in RFC-1907 that are current in RFC-1213 MUST be implemented.

4.2.2. Requirements for RFC-2233

The interface group provides essential information about both MAC interfaces and individual channels and MUST be implemented. The ifXTable SHOULD be supported.

4.2.2.1. Interface Organization and Numbering

An instance of ifEntry MUST exist for each CATV-MAC interface, downstream channel, upstream channel, and each LAN interface enabled by the CM. The enablements of LAN interfaces MAY be fixed during the manufacturing process, or MAY be determined dynamically during operation by the CM according to whether an interface has a CPE device attached to it or not.

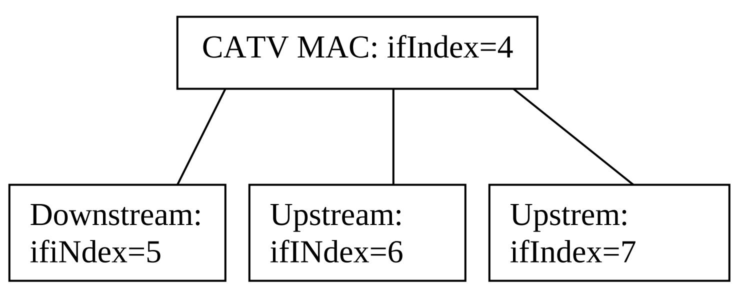
If the CM has multiple CPE interfaces but only one CPE interface can be enabled at any given time, then the ifTable MUST only contain the entry corresponding to the enabled or the default CPE interface.

If a MAC interface consists of more than one upstream and downstream channel, then a separate instance of ifEntry MUST also exist for each channel.

The ifStack group ([RFC-2233][6]) MUST be implemented to identify relationships among sub-interfaces.

Note that the CATV-MAC interface must exist, even though it is broken out into sub-interfaces.

The example below illustrates a MAC interface with one downstream and two upstream channels:



Implementation of ifStackTable for this example.

[6] ECN OSS-N-99090: Updates "RFC-1573" to "RFC-2233"

IfStackHiguerLayer	ifStackLower
4	5
4	6
4	7

Figure 4-1. Interface Numbering Example

At the CMTS, interface numbering is at the discretion of the vendor, and should correspond to the physical arrangement of connections. If table entries exist separately for upstream and downstream channels, then the ifStack group ([RFC-2233]7[7]) must be implemented to identify relationships among subinterfaces.

Note that the CATV MAC interface(s) must exist, even if further broken out into subinterfaces.

At the CM, Interfaces MUST be numbered as:

- CMCI: 1

- CATV-MAC: 2

- RF Downstream: 3

- RF Upstream: 4

- all others (additional channels, CPE ports, and telephony return interface if present): n+4

Creation of entries in both the docsDevNmAccessTable and docsDevFilter group relies on knowledge of CM interface numbering assignments. To support creation of these CM entries via configuration file SNMP encoding at registration time [RFI-SPEC], the numbering of these interfaces MUST be as described above.

The CMCI is a generic reference to any current or future form of CM CPE interface port technology [DOCSIS4]. Examples include ethernet, USB, IEEE-1394.

To determine the numbering and relationship of the remaining CM interfaces, the ifType and the ifStack table MUST be Supported.

[7] ECN OSS-N-99090: Updates “RFC-1573” to “RFC-2233”

4.2.2.2. Specific Interface Attributes

The ifAdminStatus object provides administrative control over both MAC interfaces and individual channels.

For CATV MAC interfaces, ifSpeed is defined as the bit rate of the highest-speed channel which is attached to this interface.

The ifSpecific object must be set to { docsIfMib } for CATV MAC interfaces. For upstream channels, it is set to { docsIfUpstreamChannelTable }. For downstream channels, it is set to { docsIfDownstreamChannelTable }. Note that this object is deprecated in [RFC-2233].
The ifType object has been assigned the following enumerated values for each instance of a Data Over Cable Service (DOCS) interface:

- CATV MAC interface: docsCableMacLayer (127)
- CATV downstream channel: docsCableDownstream (128)
- CATV upstream channel: docsCableUpstream (129)

4.2.3. Requirements for RFC-2011

4.2.3.1. The IP Group

The IP group MUST be implemented. It does not apply to IP packets forwarded by the device as a linklayer bridge. For the CM, it apples only to the device as an IP host. At the CMTS, it applies to the device as an IP host, and as a router if IP routing is implemented.

4.2.3.2. The ICMP Group

The ICMP group MUST be implemented. See Section 2.3.4 for additional requirements.

4.2.4. Requirements for RFC-1493

In both the CM and the CMTS (if the CMTS implements transparent bridging), the Bridge MIB ([RFC-1493]) MUST be implemented to represent the bridging process.

In the CMTS that implements transparent bridging, the Bridge MIB SHOULD be used to represent information about the MAC Forwarder states.

4.2.5. Requirements for RFC-2665

The Ethernet-like MIB ([RFC-2665]) MUST be implemented if Ethernet or Fast Ethernet interfaces are present.

4.2.6. Requirements for RFC-2013

The UDP group in [RFC-2013] MUST be implemented.

4.2.7. Requirements for RFC-1512

The FDDI MIB ([RFC-1512]) MUST be implemented if a Fiber Distributed Data Interface is present.

4.3. MIB Transition [8]

In order to help operators make a smooth MIB transition, CMs:

• MUST support the configuration file with CD-4 MIB OID and the configuration file with [RFC-2669] MIB OID

• MAY support querying of both MIBs (CD-4 and [RFC-2669]) from NMS

[8] ECN OSS-N-00011: Added Section 4.3

Appendix I Protocol Definition for SYSLOG (normative)

This appendix documents the usage of the SYSLOG protocol for the Operations Support System environment. The SYSLOG protocol is a UDP-based protocol that permits remote logging of device messages. Messages may be associated with different facilities and multiple priorities.

The basic format of the SYSLOG packet is simple to describe. The UDP source and destination port number is 514. The UDP payload consists of a facility/priority value enclosed in angle brackets, followed by a null-terminated string. The UDP payload string normally includes an optional time-of-day stamp, an identification string, an optional PID (in square brackets), and the actual logging message.

For consistency in a multi-vendor CM environment, this appendix adds further constraints to the SYSLOG packet. The CM uses the "local0" facility in its SYSLOG messages, so that the SYSLOG server can manage CM SYSLOG messages separately from kernel, mail, news, and other generic facilities. This limits the facility/priority values to the range of 128 to 135. The actual facility/priority value depends on the urgency of the message: emergency(128), alert(129), critical(130), error(131), warning(132), notice(133), information(134), and debug(135). This appendix also constrains the UDP payload string. The time-of-day stamp SHOULD NOT be included, forcing the SYSLOG server to provide its own (consistent) timestamps for all CM SYSLOG messages. The identification string MUST be "Cablemodem", and the "optional PID" MUST be a constant vendor-specific identification label, to assist in SYSLOG server logging management.

An example of a valid SYSLOG UDP payload would be “<132>Cablemodem[VendorX]: Downloading new CM software”. This example payload might be recorded on the SYSLOG server as “Jan 12 12:56:03 24.1.1.1 Cablemodem[VendorX]: Downloading new CM software”.

Appendix II References (informative)

[DOCSIS3] DOCSIS Cable Modem Termination System - Network-Side Interface Specification SPCMTS-NSI-I01-960702, July 2, 1996.

[DOCSIS4] DOCSIS Cable Modem to Customer Premise Equipment Interface Specification SP-CMCII06-010829, August 29, 2001.

[DOCSIS7] DOCSIS Cable Modem Radio Frequency Interface Specification SP-RFI-I06-010829, August 29, 2001.

[RFC-1157] Schoffstall, M., Fedor, M., Davin, J. and Case, J., A Simple Network Management Protocol (SNMP), IETF RFC-1157, May, 1990.

[RFC-1212] K. McCloghrie and M. Rose. Concise MIB Definitions, IETF RFC-1212, March, 1991.

[RFC-1213] K. McCloghrie and M. Rose. Management Information Base for Network Management of TCP/IP-base internets: MIB-II, IETF RFC-1213, March, 1991.

[RFC-1224] L. Steinberg., Techniques for Managing Asynchronously Generated Alerts, IETF RFC-1224, May, 1991.

[RFC-1493] E. Decker, P. Langille, A. Rijsinghani, and K.McCloghrie., Definitions of Managed Objects for Bridges, IETF RFC-1493, July, 1993.

[RFC-1512] J. Case and A. Rijsinghani. FDDI Management Information Base, IETF RFC-1512,September, 1993.

[RFC-1643] F. Kastenholz. Definitions of Managed Objects for the Ethernet-like Interface Types, IETF RFC-1643, July, 1994.

[RFC-1907] J. Case, K. McCloughrie, M. Rose, and S. Waldbusser. Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2), IETF RFC-1907, January, 1996.

[RFC-2011] K. McCloughrie. Management Information Base for the Internet Protocol using SMIv2, IETF RFC-2011, November, 1996.

[RFC-2013] K. McCloughrie. SNMPv2 Management Information Base for the User Datagram Protocol using SMIv2, IETF RFC-2013, November, 1996.

[RFC-2132] S. Alexander, R. Droms. DHCP Options and BOOTP Vendor Extensions. IETF RFC-2132.March, 1997.

[RFC-2665] J. Flick, J. Johnson. Definitions of Managed Objects for the Ethernet-like Interface Types,IETF RFC-2665, August, 1999.

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

[RFC-2670] M. St. Johns. Radio Frequency (RF) Interface Management Information Base for MCNS/DOCSIS compliant RF interfaces, IETF RFC-2670, August, 1999.

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

Appendix III Glossary (informative)

American National Standards Institute (ANSI) – A U.S. standards body.

ANSI – See American National Standards Institute.

Availability – In cable television systems, availability is the long-term ratio of the actual RF channel operation time to scheduled RF channel operation time (expressed as a percent value) and is based on a bit error rate (BER) assumption.

Broadcast Addresses – A predefined destination address that denotes the set of all data network service access points.

BSS – See Business Support System.

Business Support System (BSS) – a collection of computing equipment maintaining accounting, billing, and access control for a cable modem network.

Cable Modem (CM) – A modulator-demodulator at subscriber locations intended for use in conveying data communications on a cable television system.

Cable Modem Termination System (CMTS) – Cable modem termination system, located at the cable television system headend or distribution hub, which provides complementary functionality to the cable modems to enable data connectivity to a wide-area network.

Cable Modem Termination System - Network Side Interface (CMTS-NSI) – The interface, defined in [DOCSIS3], between a CMTS and the equipment on its network side.

Cable Modem to CPE Interface (CMCI) – The interface, defined in [DOCSIS4], between a CM and CPE.

CM – See Cable Modem.

CMCI – See Cable Modem to CPE Interface.

CMTS – See Cable Modem Termination System.

CMTS-NSI – See Cable Modem Termination System - Network Side Interface.

CPE – See Customer Premise Equipment.

Customer – See End User.

Customer Premises Equipment (CPE) – Equipment at the end user's premises; MAY be provided by the end user or the service provider.

Data Link Layer – Layer 2 in the Open System Interconnection (OSI) architecture; the layer that provides services to transfer data over the transmission link between open systems.

DHCP – See Dynamic Host Configuration Protocol.

Distribution Hub – A location in a cable television network which performs the functions of a Headend for customers in its immediate area, and which receives some or all of its television program material from a Master Headend in the same metropolitan or regional area; see, for example, [DOCSIS1].

DOCSIS – Data-Over-Cable Service Interface Specification.

Downstream – In cable television, the direction of transmission from the headend to the subscriber.

Dynamic Host Configuration Protocol (DHCP) – An Internet protocol used for assigning network-layer (IP) addresses.

End User – A human being, organization, or telecommunications system that accesses the network in order to communicate via the services provided by the network.

Fiber Node – A point of interface between a fiber trunk and the coaxial distribution.

Headend – The central location on the HFC network that is responsible for injecting broadcast video and other signals in the downstream direction. See also Master Headend, Distribution Hub.

Header – Protocol control information located at the beginning of a protocol data unit.

HFC – See Hybrid Fiber/Coax (HFC) System.

High Frequency (HF) – Used in this document to refer to the entire subsplit (5-30 MHz) and extended subsplit (5-42 MHz) band used in reverse channel communications over the cable television network.

Hybrid Fiber/Coax (HFC) System – A broadband bi-directional shared-media transmission system using fiber trunks between the headend and the fiber nodes, and coaxial distribution from the fiber nodes to the customer locations.

ICMP – See Internet Control Message Protocol.

IEEE – See Institute of Electrical and Electronic Engineers.

IETF – See Internet Engineering Task Force.

Internet Control Message Protocol (ICMP) – An Internet network-layer protocol.

Institute of Electrical and Electronic Engineers (IEEE) – A voluntary organization which, among other things, sponsors standards committees and is accredited by the American National Standards Institute.

Internet Engineering Task Force (IETF) – A body responsible, among other things, for developing standards used in the Internet.

Internet Protocol (IP) – An Internet network-layer protocol.

International Organization for Standardization (ISO) – An international standards body, commonly known as the International Standards Organization.

IP – See Internet Protocol.

Layer – A subdivision of the Open System Interconnection (OSI) architecture, constituted by subsystems of the same rank

LLC – See Logical Link Control (LLC) procedure.

Local Area Network (LAN) – A non-public data network in which serial transmission is used for direct data communication among data stations located on the user's premises.

Logical Link Control (LLC) procedure – In a local area network (LAN) or a Metropolitan Area Network (MAN), that part of the protocol that governs the assembling of data link layer frames and their exchange between data stations, independent of how the transmission medium is shared.

MAC – See Media Access Control (MAC) procedure.

Master Headend – A headend which collects television program material from various sources by satellite, microwave, fiber and other means, and distributes this material to Distribution Hubs in the same metropolitan or regional area. A Master Headend MAY also perform the functions of a Distribution Hub for customers in its own immediate area; see, for example, [DOCSIS1].

Media Access Control (MAC) address – The “built-in” hardware address of a device connected to a shared medium.

Media Access Control (MAC) procedure – In a subnetwork, that part of the protocol that governs access to the transmission medium independent of the physical characteristics of the medium, but taking into account the topological aspects of the subnetworks, in order to enable the exchange of data between nodes.

MAC procedures include framing, error protection, and acquiring the right to use the underlying transmission medium.

Media Access Control (MAC) sublayer – The part of the data link layer that supports topologydependent functions and uses the services of the Physical Layer to provide services to the logical link control (LLC) sublayer.

Network Layer – Layer 3 in the Open System Interconnection (OSI) architecture; the layer that provides services to establish a path between open systems.

Network Management – The functions related to the management of data link layer and physical layer resources and their stations across the data network supported by the hybrid fiber/coax system.

Open Systems Interconnection (OSI) – A framework of ISO standards for communication between different systems made by different vendors, in which the communications process is organized into seven different categories that are placed in a layered sequence based on their relationship to the user. Each layer uses the layer immediately below it and provides a ervice to the layer above. Layers 7 through 4 deal with end-to-end communication between the message source and destination, and layers 3 through 1 deal with network functions.

Operations Support System (OSS) – The backoffice software used for configuration, performance, fault, accounting and security management.

OSI – See Open Systems Interconnection.

OSS – See Operations Support System.

PHY – See Physical (PHY) Layer.

Physical (PHY) Layer – Layer 1 in the Open System Interconnection (OSI) architecture; the layer that provides services to transmit bits or groups of bits over a transmission link between open systems and which entails electrical, mechanical and handshaking procedures.

Protocol – A set of rules and formats that determines the communication behavior of layer entities in the performance of the layer functions.

QoS – See Quality of Service.

Quality of Service (QoS) –The accumulation of the cell loss, delay, and delay variation incurred by cells belonging to a particular connection.

Radio Frequency (RF) – In cable television systems, this refers to electromagnetic signals in the range 5 to 1000 MHz.

Reverse Channel – The direction of signal flow towards the headend, away from the subscriber; equivalent to Upstream.

Request For Comments (RFC) – A technical policy document of the IETF; these documents can be accessed on the World Wide Web at http://ds.internic.net/ds/rfcindex.html.

RFC – See Request for Comments.

Simple Network Management Protocol (SNMP) – A network management protocol of the ETF.

SNMP – See Simple Network Management Protocol.

Subscriber – See End User.

Sublayer – A subdivision of a layer in the Open System Interconnection (OSI) reference model.

Subnetwork – Subnetworks are physically formed by connecting adjacent nodes with transmission links.

Subsystem – An element in a hierarchical division of an Open System that interacts directly with elements in the next higher division or the next lower division of that open system. Systems Management
– Functions in the application layer related to the management of various open systems Interconnection
(OSI) resources and their status across all layers of the OSI architecture.

TFTP – See Trivial File-Transfer Protocol.

Transmission Control Protocol (TCP) – A transport-layer Internet protocol which ensures successful end-to-end delivery of data packets without error.

Trivial File-Transfer Protocol (TFTP) – An Internet protocol for transferring files without the requirement for user names and passwords that is typically used for automatic dow4nloads of data and software.

Transmission Link – The physical unit of a subnetwork that provides the transmission connection between adjacent nodes.

Transmission Medium – The material on which information signals may be carried; e.g., optical fiber, coaxial cable, and twisted-wire pairs.

Transmission System – The interface and transmission medium through which peer physical layer entities transfer bits.

Upstream – The direction from the subscriber location toward the headend.

Appendix IV MIB Requirements[9] (normative)

RFC/MIB Doc	**Table/Group/Objects**	**CM**	**CMTS**
RFC-2011	IP Group	M	M
	IpAddrTable	M	M
	ICMP Group	M	M
RFC-2013	UDP Group	M	M M
RFC-1907	System Group	M	M
	SNMP Group	M	M
RFC-2358	dot3StatsTable	M	M
	dot3CollFrequencies	O	O
RFC-2233	IfTable	M	M
	ifXTable		
	ifInMulticastPkts	M	M
	ifInBroadcastPkts	M	M
	ifOutMulticastPkts	M	M
	ifOutBroadcastPkts	M	M
	ifHCInOctets	M	M

[9] ECN OSS-N-99090: Added Appendix D

	ifHCInUcastPkts	O	O
	ifHCInMulticastPkts	O	O
	ifHCOutOctets	O	O
	iFHCOutUcastPkts	O	O
	ifHCOutMulticastPkts	O	O
	ifHCOutBroadcastPkts	O	O
	ifLinkUpDownTrapEnable	O	O
	ifHighSpeed	M	M
	ifPromiscuousMode	M	M
	ifConnectorPresent	M	M
	ifAlias	M	M
	ifCounterDiscontinuityTime	M	M
	ifStackTable	M	M
	ifStackStatus	3	
	ifTestTable		
		O	O
RFC-1493	dot1dBase Group	M	M
	dot1StpTable	1	1
	dot1StpPortTable	1	1
	dot1dtp Objects	M	M
	dot1dTpFdbTable	M	M
	dot1dTpPortTable	M	M
	dot1dStatic Table	O	O
RFC-2669	docsDevBaseGroup	M	O
	DocDevNmAccessTable	2	O
	docsDevSoftware Obj	M	O
	docsDevServer Objects	M	Not
	docsDevEvent Objects	M	Implemented
	docsDevEvControl Table	M	O
	dosDevEventTable M O	M	O
	LLC UnmatchAction Object	M	O

	IP Default Object		O
	docsDevfilterIPTable	M	
	docsDevfilterPolicyTable	M	O
	docsDevFilterTosTable	M	O
	docsDevCPE Objects	M	O
	docsDevCpeTable	M	O
		M	Not
		M	Implemented
		M	Not
		M	Implemented
		M	
		M	
RFC-2670	docsIfDownChannelTable	M	M
	docsIfUpstreamChannelTable	M	M
	docsIfQosProfileTable	M	M
	docsIfSignalQualityTable	M	M
	docsIfCmMacTable	M	Not
	docsIfCmStatusTable	M	Implemented
	docsIfCmServiceTable	M	Not
	docsIfCmtsMacTable	Not Imp	Implemented
	docsIfCmtsStatusTable	Not Imp	Not
	docsIfCmtsCmStatusTable	Not Imp	Implemented
	docsIfCmtsServiceTable	Not Imp	M
	docsIfCmtsModulation Table	Not Imp	M
	docsIfCmtsQosProfilePermission	Not Imp	M
	docsIfCmtsMactoCmTable	Not Imp	M
			M
			O
			M

Notes:

M = Mandatory.

If M applies to a table:

- "current" objects MUST be implemented if not explicitly stated in the OSSI to not
  implement.
- "deprecated" object(s) conform to OSSI rules specified above.
- "obsolete" object(s) conform to OSSI rules specified above

If M applies to an object:

- "current" objects MUST be implemented if not explicitly stated in the OSSI to not
  implement.
- "deprecated" objects conform to OSSI rules specified above.
- "obsolete" object(s) conform to OSSI rules specified above

O = Optional. If implemented, rules specified in the Mandatory section are in effect.

1 = required if STP is implemented

2 = devices running SNMPv3 agents MUST NOT implement this table

3 = Implementation of the ifStack table is only required if separate upstream and downstream interfaces are defined in the ifTable.

Appendix V USB MIB Definition[10] (normative)

```
USB-MIB DEFINITIONS ::= BEGIN
IMPORTS
        MODULE-IDENTITY,
        OBJECT-TYPE,
        Counter32,
        Integer32,
        experimental
                FROM SNMPv2-SMI
        MODULE-COMPLIANCE,
```

[10] ECN OSS-N-99094: Adds USB MIB definition

```
        OBJECT-GROUP
                FROM SNMPv2-CONF
        TEXTUAL-CONVENTION,
        MacAddress,
        TruthValue
                FROM SNMPv2-TC
        InterfaceIndexOrZero
        FROM IF-MIB;
usbMib MODULE-IDENTITY
        LAST-UPDATED "9912210000Z" -- December 21, 1999
        ORGANIZATION "DOCSIS"
        CONTACT-INFO
                " Benjamin Dolnik  Postal: 3Com Corporation  3800 Golf Road Rolling
                eadows, IL 60008 USA Phone: +1 847 262 2098 E-mail:
                benjamin_dolnik@3com.com"
DESCRIPTION
        "The MIB module to describe the USB interface."
REVISION "9911030000Z"
DESCRIPTION
        "Initial Compilable Version."
REVISION "9911100000Z"
DESCRIPTION
        "Put different CDC subclasses into separate structures, Use experimental
        group instead of transmission. Clarify descriptions"
REVISION "9912210000Z"
DESCRIPTION
        "Clean the MIB syntax. Replace some INTEGER to Integer32. Compliance
        statements added"
::= { experimental 103 } -- This number is requested but not assigned
yet
-- Generic information
usbMibObjects OBJECT IDENTIFIER ::= { usbMib 1 }
```

```
usbNumber OBJECT-TYPE
        SYNTAX          Integer32 (0..65535)
        MAX-ACCESS   read-only
        STATUS          current
        DESCRIPTION
                "The number of ports regardless of their current state in the usb
                general port table"
                ::= { usbMibObjects 1 }
--
-- usb Generic Port Table
--
usbPortTable OBJECT-TYPE
        SYNTAX SEQUENCE OF UsbPortEntry
        MAX-ACCESS          not-accessible
        STATUS                  current
        DESCRIPTION
                "A list of port entries. The number of entries is given by the value
                usbNumber."
        ::= { usbMibObjects 2 }
usbPortEntry OBJECT-TYPE
        SYNTAX          UsbPortEntry
        MAX-ACCESS not-accessible
        STATUS          current
        DESCRIPTION
                "Status and parameter values for the USB port."
        INDEX { usbPortIndex }
        ::= { usbPortTable 1 }
UsbPortEntry ::=
        SEQUENCE {
                usbPortIndex
                Integer32,
                usbPortType
```

```
            INTEGER,
            usbPortRate
            INTEGER
        }
usbPortIndex OBJECT-TYPE
        SYNTAX          Integer32 (1..65535)
        MAX-ACCESS  read-only
        STATUS          current
        DESCRIPTION
            "The unique identifier of the USB port hardware. By convention
            and if possible, hardware port numbers map directly to external
            connectors."
        ::= { usbPortEntry 1 }
usbPortType OBJECT-TYPE
        SYNTAX INTEGER {host(1), device(2), hub(3)}
        MAX-ACCESS        read-only
        STATUS               current
        DESCRIPTION
            "The type of the USB port"
        ::= { usbPortEntry 2 }
usbPortRate OBJECT-TYPE
        SYNTAX INTEGER {low-speed(1), full-speed(2), high-speed(3)}
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "The USB port rate that could be low-speed(1) for 1.5 Mbps,
            fullspeed(2) for 12Mbps or high-speed(3) for USB 2.0"
        ::= { usbPortEntry 3 }
--
-- usb Device MIB
--
usbDeviceTable OBJECT-TYPE
```

SYNTAX SEQUENCE OF UsbDeviceEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"A list of USB device ports. Usually the device has only one USB device port"
::= { usbMibObjects 3 }
usbDeviceEntry OBJECT-TYPE
SYNTAX UsbDeviceEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"Status and parameter values for the USB device port."
INDEX { usbDeviceIndex }
::= { usbDeviceTable 1 }
UsbDeviceEntry ::=
SEQUENCE {
usbDeviceIndex
Integer32,
usbDevicePower INTEGER,
usbDeviceVendorID
OCTET STRING,
usbDeviceProductID
OCTET STRING,
usbDeviceNumberConfigurations
Integer32,
usbDeviceActiveClass
INTEGER,
usbDeviceStatus
INTEGER,
usbDeviceEnumCounter
Counter32,

```
        usbDeviceRemoteWakeup
        TruthValue,
        usbDeviceRemoteWakeupOn TruthValue
      }
usbDeviceIndex OBJECT-TYPE
    SYNTAX        Integer32 (1..65535)
    MAX-ACCESS  read-only
    STATUS        current
    DESCRIPTION
        "The index is identical to usbPortIndex for the correspondent USB
        port"
    ::= { usbDeviceEntry 1 }
usbDevicePower OBJECT-TYPE
    SYNTAX        INTEGER {unknown(1),self-powered(2),bus-powered(3)}
    MAX-ACCESS  read-only
    STATUS        current
    DESCRIPTION
        "the way USB device port is powered"
    ::= { usbDeviceEntry 2 }
usbDeviceVendorID OBJECT-TYPE
SYNTAX OCTET STRING
    MAX-ACCESS  read-only
    STATUS        current
    DESCRIPTION
    "The USB device port vendor HEX-formatted string as it is provided to the
    USB host by the USB device"
    ::= { usbDeviceEntry 3 }
usbDeviceProductID OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
```

"The product ID HEX-formatted string as it is provided to the USB host by the USB device"

::= { usbDeviceEntry 4 }

usbDeviceNumberConfigurations OBJECT-TYPE

SYNTAX Integer32 (1..65535)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of configurations the USB port supports. Device port should support at least one configuration"

::= { usbDeviceEntry 5 }

usbDeviceActiveClass OBJECT-TYPE

SYNTAX INTEGER {other(1), cdc(2)}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object returns USB Device Class type of the active configuration"

::= { usbDeviceEntry 6 }

usbDeviceStatus OBJECT-TYPE

SYNTAX INTEGER {unattached(1), attached(2), powered(3), default(4), address(5), configured(6), suspended(7)}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Current status of the USB device state machine"

::= { usbDeviceEntry 7 }

usbDeviceEnumCounter OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

```
            "Total number reconnections (enumerations) since device is
            operational"
        ::= { usbDeviceEntry 8 }
usbDeviceRemoteWakeup OBJECT-TYPE
        SYNTAX          TruthValue
        MAX-ACCESS  read-only
        STATUS          current
        DESCRIPTION
            "If set to true(1), the device supports Remote Wakeup function.
            If set to false(2), the device doesn't support it"
        ::= { usbDeviceEntry 9 }
usbDeviceRemoteWakeupOn OBJECT-TYPE
        SYNTAX TruthValue
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "If set to true(1), the remote wakeup function is activated by the host.
            If set to false(2), remote wakeup function is not active."
        ::= { usbDeviceEntry 10 }
        --
        -- Table of the CDC interfaces
        --
usbCDCTable OBJECT-TYPE
        SYNTAX SEQUENCE OF UsbCDCEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "A list of Communication Device Class (CDC) interfaces supported by
            the USB device. It could be more than one CDC interface for the
            device that expose more than one interface to the network"
        ::= { usbMibObjects 4 }
usbCDCEntry OBJECT-TYPE
```

```
    SYNTAX UsbCDCEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Status and parameter values for CDC device"
    INDEX { usbCDCIndex, usbCDCIfIndex }
    ::= { usbCDCTable 1 }
UsbCDCEntry ::=
    SEQUENCE {
    usbCDCIndex
    Integer32,
    usbCDCIfIndex
    InterfaceIndexOrZero,
    usbCDCSubclass
    INTEGER,
    usbCDCVersion
        OCTET STRING,
    usbCDCDataTransferType
        INTEGER,
    usbCDCDataEndpoints
        Integer32,
    usbCDCStalls
        Counter32
    }
usbCDCIndex OBJECT-TYPE
    SYNTAX Integer32 (1..65535)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The index is identical to usbPortIndex for the correspondent USB
        port"
    ::= { usbCDCEntry 1 }
```

```
usbCDCIfIndex OBJECT-TYPE
        SYNTAX InterfaceIndexOrZero
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
                "The variable uniquely identifies the interface index which this CDC
                device is representing"
        ::= { usbCDCEntry 2 }
usbCDCSubclass OBJECT-TYPE
        SYNTAX INTEGER  {other(0), directLine(1), acm(2), telephony(3),
        multichannel(4), capi(5), ethernet(6), atm(7)}
        MAX-ACCESS        read-only
        STATUS            current
        DESCRIPTION
                "Subclass used in data transfer in Communication Device Class"
        REFERENCE
        "USB Class definitions for Communication Devices ver 1.1, p.28"
        ::= { usbCDCEntry 3 }
usbCDCVersion OBJECT-TYPE
        SYNTAX OCTET STRING (SIZE (2))
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
                "String that describes the version of Communication Device Class in
                HEX format (Major, Minor) "
        ::= { usbCDCEntry 4 }
usbCDCDataTransferType OBJECT-TYPE
        SYNTAX            INTEGER {synchronous(1), asynchronous(2)}
        MAX-ACCESS        read-only
        STATUS            current
        DESCRIPTION
```

```
        "Type of data transfer for Data Class Interface used by the
        Communication Device. Isochronous mode is used for synchronous(1)
        and bulk transfer mode is used for asynchronous(2)"
    ::= { usbCDCEntry 5 }
usbCDCDataEndpoints OBJECT-TYPE
    SYNTAX         Integer32 (0..16)
    MAX-ACCESS  read-only
    STATUS         current
    DESCRIPTION
        "Number of the data endpoints (IN and OUT) used by the
        Communication Device. If the networking device is in default interface
        setting, there are are no.data endpoints and no traffic is exchanged.
        Under the normal operation there should be 2 Data Endpoints (one IN
        and one OUT) for the networking device.
        For the multichannel model this number could be larger than 2."
    ::= { usbCDCEntry 6 }
usbCDCStalls OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Total number of times USB Data interface recovered from stall since
        re-initialization and while the port state was 'up' or 'test'."
    ::= { usbCDCEntry 7 }
    --
    -- Table of the CDC Ethernet-type interfaces
    --
usbCDCEtherTable OBJECT-TYPE
    SYNTAX SEQUENCE OF UsbCDCEtherEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
```

```
        "A list of Communication Device Class (CDC) USB devices that
        support Ethernet Networking Control Model."
    ::= { usbMibObjects 5 }
usbCDCEtherEntry OBJECT-TYPE
    SYNTAX UsbCDCEtherEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Status and parameter values for CDC devices that support Ethernet
        Networking Control Model"
    INDEX { usbCDCEtherIndex, usbCDCEtherIfIndex }
    ::= { usbCDCEtherTable 1 }
UsbCDCEtherEntry ::=
    SEQUENCE {
    usbCDCEtherIndex
    Integer32,
    usbCDCEtherIfIndex
    InterfaceIndexOrZero,
usbCDCEtherMacAddress
    MacAddress,
    usbCDCEtherPacketFilter
    BITS,
    usbCDCEtherDataStatisticsCapabilities
    BITS,
usbCDCEtherDataCheckErrs
    Counter32
}
usbCDCEtherIndex OBJECT-TYPE
    SYNTAX Integer32 (1..65535)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
```

```
        "The index is identical to usbPortIndex for the correspondent USB
        port"
    ::= { usbCDCEtherEntry 1 }
usbCDCEtherIfIndex OBJECT-TYPE
    SYNTAX InterfaceIndexOrZero
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The variable uniquely identifies the interface index to which
        this CDC device is connected "
    ::= { usbCDCEtherEntry 2 }
usbCDCEtherMacAddress OBJECT-TYPE
    SYNTAX       MacAddress
    MAX-ACCESS read-only
    STATUS      current
    DESCRIPTION
        "The 48bit MAC address that is provided by USB CDC device to the
        host. This address will be used as the source address of Ethernet
        frames sent by the host over the particular CDC interface."
    ::= { usbCDCEtherEntry 3}
usbCDCEtherPacketFilter OBJECT-TYPE
    SYNTAX BITS {
        packetPromiscuous(0),
        packetAllMulticast(1),
        packetDirected(2),
        packetBroadcast(3),
        packetMulticast(4)
        }
    MAX-ACCESS   read-only
    STATUS        current
    DESCRIPTION
```

"Bitmap indicates the host requirements to the USB device to perform Ethernet packet filtering of the particular type frames directed to the host."

REFERENCE

"USB Class definitions for Communication Devices ver 1.1, p.66 Table 62"

::= { usbCDCEtherEntry 4 }

usbCDCEtherDataStatisticsCapabilities OBJECT-TYPE

SYNTAX BITS {

frameXmitOk(0),

frameRcvOk(1),

frameXmitErr(2),

frameRcvErr(3),

frameRcvNoBuff(4),

bytesXmitDirectOk(5),

framesXmitDirectOk(6),

bytesXmitMulticastOk(7),

framesXmitMulticastOk(8),

bytesXmitBroadcastOk(9),

framesXmitBroadcastOk(10),

bytesRcvDirectOk(11),

framesRcvDirectOk(12),

bytesRcvMulticastOk(13),

framesRcvMulticastOk(14),

bytesRcvBroadcastOk(15),

framesRcvBroadcastOk(16),

framesRcvCrcErr(17),

xmitQueueLen(18),

rcvErrAlignment(19),

xmitOneCollision(20),

xmitMoreCollisions(21),

xmitDeferred(22),

xmitMaxCollision(23),

```
        rcvOverrun(24),
        xmitUnderrun(25),
        xmitHearbeatFailure(26),
        xmitTimesCrsLost(27),
        xmitLateCollisions(28)
        }
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                "Bitmap indicates the ability to collect Ethernet statistics of different
                types as it provided in Ethernet Networking Functional Descriptor. If
                the particular bit is set, the device could provide the corresponding
                statistics counter to the host"
        REFERENCE
        "USB Class definitions for Communication Devices ver 1.1, p.46 Table 42"
        ::= { usbCDCEtherEntry 5 }
usbCDCEtherDataCheckErrs OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
                "Total number of frames with an invalid frame check sequence, input
                from the USB Data interface since system re-initialization and while
                the port state was 'up' or 'test'."
        ::= { usbCDCEtherEntry 6 }
        --
        -- notification group is for future extension.
        --
usbMibNotification OBJECT IDENTIFIER ::= { usbMib 2 }
usbMibConformance OBJECT IDENTIFIER ::= { usbMib 3 }
usbMibCompliances OBJECT IDENTIFIER ::= { usbMibConformance 1 }
usbMibGroups OBJECT IDENTIFIER ::= { usbMibConformance 2 }
```

```
-- compliance statements
usbMibBasicCompliance MODULE-COMPLIANCE
        STATUS current
        DESCRIPTION
                "The compliance statement for devices that implement USB MIB"
        MODULE -- usbMib
        -- unconditionally mandatory groups
        MANDATORY-GROUPS {
                        usbMibBasicGroup
             }
-- conditionally mandatory group
GROUP usbMibCDCGroup
        DESCRIPTION
                "This group is implemented only in devices having at least one CDC
                interface"
        -- conditionally mandatory group
  GROUP usbMibCDCEtherGroup
DESCRIPTION
        "This group is implemented only in devices supporting at least one CDC
        interface that uses Ethernet Networking Control Model"
::= {usbMibCompliances 1}
usbMibBasicGroup OBJECT-GROUP
OBJECTS {
        usbNumber,
        usbPortIndex,
        usbPortType,
        usbPortRate,
        usbDeviceIndex,
        usbDevicePower,
        usbDeviceVendorID,
        usbDeviceProductID,
        usbDeviceNumberConfigurations,
```

```
        usbDeviceActiveClass,
        usbDeviceStatus,
        usbDeviceEnumCounter,
        usbDeviceRemoteWakeup,
        usbDeviceRemoteWakeupOn
 }
STATUS current
        DESCRIPTION
        "Group of objects that are mandatory to support by device implementing this
        MIB"
        ::= { docsIfGroups 1 }
        usbMibCDCGroup OBJECT-GROUP
        OBJECTS {
        usbCDCIndex,
        usbCDCIfIndex,
        usbCDCSubclass,
        usbCDCVersion,
usbCDCDataTransferType,
usbCDCDataEndpoints,
usbCDCStalls
}
STATUS current
        DESCRIPTION
                "This group is implemented only in devices having at least one CDC
                interface"
        ::= { docsIfGroups 2 }
usbMibCDCEtherGroup OBJECT-GROUP
        OBJECTS {
                usbCDCEtherIndex,
                usbCDCEtherIfIndex,
                usbCDCEtherMacAddress,
                usbCDCEtherPacketFilter,
```

```
            usbCDCEtherDataStatisticsCapabilities,
            usbCDCEtherDataCheckErrs
    }
STATUS current
DESCRIPTION
    "This group is implemented only in devices having at least one CDC interface
    that uses Ethernet Networking Control Model"
::= { docsIfGroups 3 }
END
```

Appendix VI Revision History (Informative)

VI.1 ECNs Included in SP-OSSI-RFI-I04-010829

Table VI.1. ECN Review Table

ECN	Date Accepted
OSS-N-99020	05/06/99
OSS-N-99031	05/26/99
OSS-N-99082	10/06/99
OSS-N-99090	11/24/99
OSS-N-99094	01/05/00
OSS-N-00011	03/29/00
OSS-N-01034	05/09/01

Data-Over-Cable Service Interface Specifications

Operations Support System Interface Specification

Radio Frequency Interface

SP-OSSI-RFI-I04-010829

INTERIM

SPECIFICATION

Document Status Sheet

Document Control Number:	SP-OSSI-RFI-I04-010829
Document Title:	Operations Support System Interface Specification Radio Frequency Interface
Revision History:	
Date:	08/29/01
Status:	~~Work in Progress~~ ~~Draft~~ Interim ~~Released~~
Distribution Restrictions:	~~Author Only~~ ~~CL/Member~~ ~~CL/Member/ Vendor~~ Public

Key to Document Status Codes:

Work in Progress An incomplete document, designed to guide discussion and generate feedback, that may include several alternative requirements for consideration.

Draft A document in specification format considered largely complete, but lacking review by Members and vendors. Drafts are susceptible to substantial change during the review process.

Interim A document which has undergone rigorous Member and vendor review, suitable for use by vendors to design in conformance to and for field testing.

For purposes of the "Contribution and License Agreement for Intellectual Property" which grants licenses to the intellectual property contained in the DOCSIS Specification, an "Interim" Specification is a "Published" Specification.

Released A stable document, reviewed, tested and validated, suitable to enable cross-vendor interoperability.

CONTENTS