Definitions of Managed Objects for
the Universal Serial Bus (USB) Interface

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026 [18].
Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts. Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt
The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

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 objects for managing Universal Serial Bus (USB) interfaces.

Table of Contents

1. Conventions used in this document........................................1
2. The SNMP Management Framework........................................2
3. Glossary...............................................................................2
4. Overview............................................................................3
4.1. Structure of the MIB....................................................3
4.2. Relationship to the Interfaces MIB....................................3
5. Definitions...........................................................................3
6. Security Considerations....................................................4
7. References...........................................................................17
8. Author's Addresses..........................................................17
9. Full Copyright Statement..................................................20

1. Conventions used in this document

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 [19].
The SNMP Management Framework presently consists of five major components:

- An overall architecture, described in RFC 2571 [1].
- Mechanisms for describing and naming objects and events for the purpose of management. The first version of this Structure of Management Information (SMI) is called SMIv1 and described in STD 16, RFC 1155 [2], STD 16, RFC 1212 [3] and RFC 1215 [4]. The second version, called SMIv2, is described in STD 58, RFC 2578 [5], STD 58, RFC 2579 [6] and STD 58, RFC 2580 [7].
- Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in STD 15, RFC 1157 [8]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in RFC 1901 [9] and RFC 1906 [10]. The third version of the message protocol is called SNMPv3 and described in RFC 1906 [10], RFC 2572 [11] and RFC 2574 [12].
- Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in STD 15, RFC 1157 [8]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [13].
- A set of fundamental applications described in RFC 2573 [14] and the view-based access control mechanism described in RFC 2575 [15]. A more detailed introduction to the current SNMP Management Framework can be found in RFC 2570 [RFC2570].

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

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

3. Glossary

USB
Universal Serial Bus

B. Dolnik
Experimental

Draft-dolnik-usb-mib-02 USB interface MIB August 2000

CDC
Communication Device Class
4. Overview

This MIB contains set of objects required for management of the USB interface. This specification is derived in part from the parameters described in the document "USB Class definitions for Communication Devices"[24]. In Current status this MIB is specifically describes 2 USB interface subclasses of the Communication Device Class Abstract Control Model, used by Remote NDIS[23] and Ethernet Networking Control Model. It could be extended in the future to specify other type of USB classes and subclasses.

4.1. Structure of the MIB

The MIB consist of the mandatory usbMibBasicGroup, describing the physical USB port (one table entry per physical port) and conditionally-mandatory groups usbMibCDCGroup and usbMibCDCEtherGroup. The last 2 groups describe virtual connections for the particular physical port. Because physical port theoretically may support more then one CDC connection, it could be more then one UsbCDCEtherEntry and/or UsbCDCEntry entry per physical port. The additional table, ifCDCEtherXmtAddressTable is conditionally-mandatory for the CDC connections that support transmit frame filtering based on destination address.

4.2. Relationship to the Interfaces MIB

This section clarifies the relation of this MIB to the Interfaces MIB[17]. Each USB interface of one of the CDC subclasses MUST have the entry in the Interfaces MIB with the ifIndex identical to usbCDCIfIndex.

Layering Model

This MIB doesn’t specify the layering model and don’t support sublayers.

Virtual Circuits

Every CDC interface of the particular USB port is a separate virtual circuit and MUST have its own entry in the ifTable.

ifRcvAddressTable

ifRcvAddressTable is not a requirement for this MIB. If the USB interface supports filtering for the outgoing traffic transmitted to the host, the usbCDCEtherXmtAddressTable MUST be supported.

ifPhysAddress

For the USB interface which usbCDCSubclass is Ethernet or ACM and it uses Remote NDIS over the Abstract Control Model, ifPhysAddress contains the IEEE 802.3 address which is placed in the source-address field of Ethernet frames which are transferred through this interface.

B. Dolnik                    Experimental                            3
Draft-dolnik-usb-mib-02   USB interface MIB               August 2000

ifType

ifType of the USB interface MUST have the IANA value of usb (160).

ifSpeed
ifSpeed for this interface returns maximum raw bandwidth in bits/s supported by the USB port. For the full-speed interface this value is 12000000.

ifSpecific
For agents which implement the deprecated ifSpecific object, an instance of this object that is associated with USB interface MUST have the OBJECT IDENTIFIER value:

```
usbMib OBJECT IDENTIFIER ::= {experimental 130}
```

ifConnectorPresent
ifConnectorPresent will normally be "true"

5. Definitions

```
USB-MIB DEFINITIONS ::= BEGIN
IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, Counter32, Integer32,
    experimental
    FROM SNMPv2-SMI
    MODULE-COMPLIANCE, OBJECT-GROUP
    FROM SNMPv2-CONF
    TEXTUAL-CONVENTION, MacAddress, TruthValue
    FROM SNMPv2-TC
    InterfaceIndexOrZero
    FROM IF-MIB;

usbMib MODULE-IDENTITY
    LAST-UPDATED    "200008070000Z"  -- August 07, 2000
    ORGANIZATION    "3Com"
    CONTACT-INFO
        "Benjamin Dolnik
        Postal: 3Com Corporation
        3800 Golf Road
        Rolling Meadows, IL 60008
        USA
        Phone: +1 847 262 2098
        E-mail: benjamin_dolnik@3com.com"
    DESCRIPTION
        "The MIB module to describe the USB interface."
    ::= { experimental 103 }

B. Dolnik                Experimental                            4
Draft-dolnik-usb-mib-02   USB interface MIB               August 2000

-- 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 }  

B. Dolnik                    Experimental                            5  
Draft-dolnik-usb-mib-02   USB interface MIB               August 2000  

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 Mbp,
   full-speed(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

B. Doinik Experimental 7
Draft-dolnik-usb-mib-02 USB interface MIB August 2000

STATUS current
DESCRIPTION "The product ID HEX-formatted string as it is provided to the USB host by the USB device"
::= { usbDeviceEntry 4 }

usbDeviceNumberOfConfigurations 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 }

B. Dolnik                    Experimental                            8
Draft-dolnik-usb-mib-02   USB interface MIB               August 2000

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 then one CDC interface for the device that expose more then 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 }

B. Dolnik                Experimental   9

Draft-dolnik-usb-mib-02   USB interface MIB     August 2000

OCTET STRING,
usbCDCDataTransferType
    INTEGER,
usbCDCDataEndpoints
    Integer32,
usbCDCStalls
    Counter32
}
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. Isochronious 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 then 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 interface or interface that uses
-- Remote NDIS over Abstract Control Model
--

usbCDCEtherTable OBJECT-TYPE
SYNTAX SEQUENCE OF UsbCDCEtherEntry
MAX-ACCESS not-accessible

B. Dolnik                    Experimental                           11
Draft-dolnik-usb-mib-02    USB interface MIB                    August 2000

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

B. Dolnik                    Experimental                           12

Draft-dolnik-usb-mib-02    USB interface MIB               August 2000

   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"
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 }
"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 }

usbCDCEtherXmtAddressTable OBJECT-TYPE
SYNTAX      SEQUENCE OF UsbCDCEtherXmtAddressEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
"This table contains an entry for each multicast address for which the system will transmit packets/frames on a particular USB interface."

::= { usbMibObjects 6 }

usbCDCEtherXmtAddressEntry OBJECT-TYPE
SYNTAX      UsbCDCEtherXmtAddressEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
"A list of objects identifying an address for which the system will send packets/frames on the particular USB interface identified by the index values usbCDCIndex and ifIndex."
INDEX     {usbCDCEtherIndex, usbCDCEtherIfIndex, ifCDCEtherXmtAddress }

::= { usbCDCEtherXmtAddressTable 1 }

UsbCDCEtherXmtAddressEntry ::= 
SEQUENCE { 
     ifCDCEtherXmtAddress MacAddress
}

ifCDCEtherXmtAddress OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS read-only
STATUS      current
DESCRIPTION
"An address for which the system will will send packets/frames on the particular USB interface. The address only could be set by the host by using the command for USB interface."

::= { usbCDCEtherXmtAddressEntry 1 }

-- 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
}

-- unconditionally mandatory group
GROUP        usbMibBasicGroup
DESCRIPTION  "Group of objects that are mandatory to support by
device implementing this MIB"

-- 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 having at
least one CDC interface that uses Ethernet Networking
Control Model or remote NDIS"

-- conditionally mandatory group
GROUP        usbCDCEtherXmtAddressGroup
DESCRIPTION  "This group is implemented only for USB CDC interfaces
that have transmit multicast filtering capabilities."

::= {usbMibCompliances 1}

usbMibBasicGroup OBJECT-GROUP
OBJECTS {
  usbNumber,
  usbPortIndex,
  usbPortType,
  usbPortRate,
  usbDeviceIndex,
  usbDevicePower,
  usbDeviceVendorID,
  usbDeviceProductID,
  usbDeviceNumberConfigurations,
  usbDeviceActiveClass,
  usbDeviceStatus,
  usbDeviceEnumCounter,
  usbDeviceRemoteWakeup,
usbDeviceRemoteWakeOn
}

usbMibCDCGroup OBJECT-GROUP
OBJECTS {
  usbCDCIndex,
  usbCDCIfIndex,
  usbCDCSubclass,
  usbCDCVersion,
  usbCDCDataTransferType,
  usbCDCDataEndpoints,
  usbCDCStalls
}

usbMibCDCEtherGroup OBJECT-GROUP
OBJECTS {
  usbCDCEtherIndex,
  usbCDCEtherIfIndex,
  usbCDCEtherMacAddress,
  usbCDCEtherPacketFilter,
  usbCDCEtherDataStatisticsCapabilities,
  usbCDCEtherDataCheckErrs
}

usbCDCEtherXmtAddressGroup OBJECT-GROUP
OBJECTS {
  ifCDCEtherXmtAddress
}

END

6. Security Considerations
This MIB contains readable objects whose values provide the number and status of a device’s network interface of the USB type. There are no management objects defined in this MIB that have a MAX-ACCESS clause of read-write and/or read-create. So, if this MIB is implemented correctly, then there is no risk that an intruder can alter or create any management objects of this MIB via direct SNMP SET operations.

There are a number of managed objects in this MIB that may be considered to contain sensitive information. Therefore, it may be important in some environments to control read access to these objects and possibly to even encrypt the values of these object when sending them over the network via SNMP. It is recommended that the implementers consider the security features as provided by the SNMPv3 framework. Specifically, the use of the User-based Security Model RFC2274[20] and the View-based Access Control Model RFC2275[21] is recommended.

7. References


B. Dolnik                    Experimental                           18
Draft-dolnik-usb-mib-02    USB interface MIB               August 2000


[23] Remote NDIS Specification. Microsoft Corporation. Revision 0.80 October 1999


8. Author’s Addresses

Benjamin Dolnik
3Com Corporation
3800 Golf road
Rolling Meadows, IL
60008
Email: benjamin_dolnik@3com.com
9. Full Copyright Statement

"Copyright (C) The Internet Society (2000). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE."