```
UDP-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, Integer32, Counter32, Counter64,
    Unsigned32, IpAddress, mib-2          FROM SNMPv2-SMI
    MODULE-COMPLIANCE, OBJECT-GROUP       FROM SNMPv2-CONF
    InetAddress, InetAddressType,
    InetPortNumber                        FROM INET-ADDRESS-MIB;

udpMIB MODULE-IDENTITY
    LAST-UPDATED "200505200000Z"    -- May 20, 2005
    ORGANIZATION
        "IETF IPv6 Working Group
         http://www.ietf.org/html.charters/ipv6-charter.html"
    CONTACT-INFO
        "Bill Fenner (editor)

         AT&T Labs -- Research
         75 Willow Rd.
         Menlo Park, CA 94025

         Phone: +1 650 330-7893
         Email: <fenner@research.att.com>

         John Flick (editor)
```

Hewlett-Packard Company

8000 Foothills Blvd. M/S 5557

Roseville, CA 95747

Phone: +1 916 785 4018

Email: <john.flick@hp.com>

Send comments to <ipv6@ietf.org>"

DESCRIPTION

"The MIB module for managing UDP implementations.

Copyright (C) The Internet Society (2005). This

version of this MIB module is part of RFC 4113;

see the RFC itself for full legal notices."

REVISION "200505200000Z" -- May 20, 2005

DESCRIPTION

"IP version neutral revision, incorporating the

following revisions:

- Added udpHCInDatagrams and udpHCOutDatagrams in order
  to provide high-capacity counters for fast networks.
- Added text to the descriptions of all counter objects
  to indicate how discontinuities are detected.
- Deprecated the IPv4-specific udpTable and replaced it

```
                    with the version neutral udpEndpointTable.  This
                    table includes support for connected UDP endpoints
                    and support for identification of the operating
                    system process associated with a UDP endpoint.
                  - Deprecated the udpGroup and replaced it with object
                    groups representing the current set of objects.
                  - Deprecated udpMIBCompliance and replaced it with
                    udpMIBCompliance2, which includes the compliance
                    information for the new object groups.

                  This version published as RFC 4113."
      REVISION        "199411010000Z"       -- November 1, 1994
      DESCRIPTION
              "Initial SMIv2 version, published as RFC 2013."
      REVISION        "199103310000Z"       -- March 31, 1991
      DESCRIPTION
              "The initial revision of this MIB module was part of
               MIB-II, published as RFC 1213."
      ::= { mib-2 50 }

-- the UDP group

udp         OBJECT IDENTIFIER ::= { mib-2 7 }

udpInDatagrams OBJECT-TYPE
      SYNTAX        Counter32
```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of UDP datagrams delivered to UDP users.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by discontinuities in the value of sysUpTime."

::= { udp 1 }

udpNoPorts OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of received UDP datagrams for which there was no application at the destination port.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at

other times as indicated by discontinuities in the
value of sysUpTime."

::= { udp 2 }

udpInErrors OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of received UDP datagrams that could not be
delivered for reasons other than the lack of an
application at the destination port.

Discontinuities in the value of this counter can occur
at re-initialization of the management system, and at
other times as indicated by discontinuities in the
value of sysUpTime."

::= { udp 3 }

udpOutDatagrams OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of UDP datagrams sent from this
entity.

Discontinuities in the value of this counter can occur
at re-initialization of the management system, and at
other times as indicated by discontinuities in the
value of sysUpTime."

::= { udp 4 }

udpHCInDatagrams OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of UDP datagrams delivered to UDP
users, for devices that can receive more than 1
million UDP datagrams per second.

Discontinuities in the value of this counter can occur
at re-initialization of the management system, and at
other times as indicated by discontinuities in the
value of sysUpTime."

::= { udp 8 }

udpHCOutDatagrams OBJECT-TYPE

SYNTAX Counter64

```
        MAX-ACCESS read-only
        STATUS       current
        DESCRIPTION
                "The total number of UDP datagrams sent from this
                 entity, for devices that can transmit more than 1
                 million UDP datagrams per second.

                 Discontinuities in the value of this counter can occur
                 at re-initialization of the management system, and at
                 other times as indicated by discontinuities in the
                 value of sysUpTime."
        ::= { udp 9 }

--
-- { udp 6 } was defined as the ipv6UdpTable in RFC2454's
-- IPV6-UDP-MIB.   This RFC obsoletes RFC 2454, so { udp 6 } is
-- obsoleted.
--

-- The UDP "Endpoint" table.

udpEndpointTable OBJECT-TYPE
        SYNTAX       SEQUENCE OF UdpEndpointEntry
        MAX-ACCESS not-accessible
        STATUS       current
        DESCRIPTION
```

"A table containing information about this entity's UDP endpoints on which a local application is currently accepting or sending datagrams.

The address type in this table represents the address type used for the communication, irrespective of the higher-layer abstraction. For example, an application using IPv6 'sockets' to communicate via IPv4 between ::ffff:10.0.0.1 and ::ffff:10.0.0.2 would use InetAddressType ipv4(1).

Unlike the udpTable in RFC 2013, this table also allows the representation of an application that completely specifies both local and remote addresses and ports. A listening application is represented in three possible ways:

1) An application that is willing to accept both IPv4 and IPv6 datagrams is represented by a udpEndpointLocalAddressType of unknown(0) and a udpEndpointLocalAddress of ''h (a zero-length octet-string).

2) An application that is willing to accept only IPv4
or only IPv6 datagrams is represented by a
udpEndpointLocalAddressType of the appropriate
address type and a udpEndpointLocalAddress of
'0.0.0.0' or '::' respectively.

3) An application that is listening for datagrams only
for a specific IP address but from any remote
system is represented by a
udpEndpointLocalAddressType of the appropriate
address type, with udpEndpointLocalAddress
specifying the local address.

In all cases where the remote is a wildcard, the
udpEndpointRemoteAddressType is unknown(0), the
udpEndpointRemoteAddress is ''h (a zero-length
octet-string), and the udpEndpointRemotePort is 0.

If the operating system is demultiplexing UDP packets
by remote address and port, or if the application has
'connected' the socket specifying a default remote
address and port, the udpEndpointRemote* values should
be used to reflect this."

::= { udp 7 }

```
udpEndpointEntry OBJECT-TYPE
    SYNTAX      UdpEndpointEntry
    MAX-ACCESS not-accessible
    STATUS      current

    DESCRIPTION
            "Information about a particular current UDP endpoint.

             Implementers need to be aware that if the total number
             of elements (octets or sub-identifiers) in
             udpEndpointLocalAddress and udpEndpointRemoteAddress
             exceeds 111, then OIDs of column instances in this table
             will have more than 128 sub-identifiers and cannot be
             accessed using SNMPv1, SNMPv2c, or SNMPv3."
    INDEX   { udpEndpointLocalAddressType,
              udpEndpointLocalAddress,
              udpEndpointLocalPort,
              udpEndpointRemoteAddressType,
              udpEndpointRemoteAddress,
              udpEndpointRemotePort,
              udpEndpointInstance }
    ::= { udpEndpointTable 1 }

UdpEndpointEntry ::= SEQUENCE {
```

```
        udpEndpointLocalAddressType      InetAddressType,
        udpEndpointLocalAddress          InetAddress,
        udpEndpointLocalPort             InetPortNumber,
        udpEndpointRemoteAddressType     InetAddressType,
        udpEndpointRemoteAddress         InetAddress,
        udpEndpointRemotePort            InetPortNumber,
        udpEndpointInstance              Unsigned32,
        udpEndpointProcess               Unsigned32
    }

udpEndpointLocalAddressType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS not-accessible
    STATUS      current
    DESCRIPTION
            "The address type of udpEndpointLocalAddress.   Only
             IPv4, IPv4z, IPv6, and IPv6z addresses are expected, or
             unknown(0) if datagrams for all local IP addresses are
             accepted."
    ::= { udpEndpointEntry 1 }

udpEndpointLocalAddress OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS not-accessible
    STATUS      current
    DESCRIPTION
```

"The local IP address for this UDP endpoint.

The value of this object can be represented in three

possible ways, depending on the characteristics of the listening application:

1. For an application that is willing to accept both IPv4 and IPv6 datagrams, the value of this object must be ''h (a zero-length octet-string), with the value of the corresponding instance of the udpEndpointLocalAddressType object being unknown(0).

2. For an application that is willing to accept only IPv4 or only IPv6 datagrams, the value of this object must be '0.0.0.0' or '::', respectively, while the corresponding instance of the udpEndpointLocalAddressType object represents the appropriate address type.

3. For an application that is listening for data destined only to a specific IP address, the value of this object is the specific IP address for which this node is receiving packets, with the

corresponding instance of the
udpEndpointLocalAddressType object representing the
appropriate address type.

As this object is used in the index for the
udpEndpointTable, implementors of this table should be
careful not to create entries that would result in OIDs
with more than 128 subidentifiers; else the information
cannot be accessed using SNMPv1, SNMPv2c, or SNMPv3."
::= { udpEndpointEntry 2 }

udpEndpointLocalPort OBJECT-TYPE
SYNTAX InetPortNumber
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"The local port number for this UDP endpoint."
::= { udpEndpointEntry 3 }

udpEndpointRemoteAddressType OBJECT-TYPE
SYNTAX InetAddressType
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"The address type of udpEndpointRemoteAddress. Only
IPv4, IPv4z, IPv6, and IPv6z addresses are expected, or

unknown(0) if datagrams for all remote IP addresses are

accepted. Also, note that some combinations of

udpEndpointLocalAdressType and

udpEndpointRemoteAddressType are not supported. In

particular, if the value of this object is not

unknown(0), it is expected to always refer to the

same IP version as udpEndpointLocalAddressType."

::= { udpEndpointEntry 4 }

udpEndpointRemoteAddress OBJECT-TYPE

SYNTAX InetAddress

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The remote IP address for this UDP endpoint. If

datagrams from any remote system are to be accepted,

this value is ''h (a zero-length octet-string).

Otherwise, it has the type described by

udpEndpointRemoteAddressType and is the address of the

remote system from which datagrams are to be accepted

(or to which all datagrams will be sent).

As this object is used in the index for the

udpEndpointTable, implementors of this table should be
careful not to create entries that would result in OIDs
with more than 128 subidentifiers; else the information
cannot be accessed using SNMPv1, SNMPv2c, or SNMPv3."
::= { udpEndpointEntry 5 }

udpEndpointRemotePort OBJECT-TYPE
SYNTAX InetPortNumber
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"The remote port number for this UDP endpoint. If
datagrams from any remote system are to be accepted,
this value is zero."
::= { udpEndpointEntry 6 }

udpEndpointInstance OBJECT-TYPE
SYNTAX Unsigned32 (1..'ffffffff'h)
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"The instance of this tuple. This object is used to
distinguish among multiple processes 'connected' to
the same UDP endpoint. For example, on a system
implementing the BSD sockets interface, this would be
used to support the SO_REUSEADDR and SO_REUSEPORT

socket options."

::= { udpEndpointEntry 7 }

udpEndpointProcess OBJECT-TYPE
SYNTAX Unsigned32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The system's process ID for the process associated with
this endpoint, or zero if there is no such process.
This value is expected to be the same as
HOST-RESOURCES-MIB::hrSWRunIndex or SYSAPPL-MIB::
sysApplElmtRunIndex for some row in the appropriate
tables."
::= { udpEndpointEntry 8 }

-- The deprecated UDP Listener table

-- The deprecated UDP listener table only contains information
-- about this entity's IPv4 UDP end-points on which a local
-- application is currently accepting datagrams. It does not
-- provide more detailed connection information, or information
-- about IPv6 endpoints.

```
udpTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF UdpEntry
    MAX-ACCESS not-accessible
    STATUS      deprecated
    DESCRIPTION
            "A table containing IPv4-specific UDP listener
             information.  It contains information about all local
             IPv4 UDP end-points on which an application is
             currently accepting datagrams.  This table has been
             deprecated in favor of the version neutral
             udpEndpointTable."
    ::= { udp 5 }

udpEntry OBJECT-TYPE
    SYNTAX      UdpEntry
    MAX-ACCESS not-accessible
    STATUS      deprecated
    DESCRIPTION
            "Information about a particular current UDP listener."
    INDEX   { udpLocalAddress, udpLocalPort }
    ::= { udpTable 1 }

UdpEntry ::= SEQUENCE {
    udpLocalAddress     IpAddress,
    udpLocalPort        Integer32
```

}

udpLocalAddress OBJECT-TYPE

SYNTAX IpAddress

MAX-ACCESS read-only

STATUS deprecated

DESCRIPTION

"The local IP address for this UDP listener. In the case of a UDP listener that is willing to accept datagrams for any IP interface associated with the node, the value 0.0.0.0 is used."

::= { udpEntry 1 }

udpLocalPort OBJECT-TYPE

SYNTAX Integer32 (0..65535)

MAX-ACCESS read-only

STATUS deprecated

DESCRIPTION

"The local port number for this UDP listener."

::= { udpEntry 2 }

-- conformance information

```
udpMIBConformance OBJECT IDENTIFIER ::= { udpMIB 2 }
udpMIBCompliances OBJECT IDENTIFIER ::= { udpMIBConformance 1 }
udpMIBGroups      OBJECT IDENTIFIER ::= { udpMIBConformance 2 }

-- compliance statements

udpMIBCompliance2 MODULE-COMPLIANCE
    STATUS      current
    DESCRIPTION
            "The compliance statement for systems that implement
             UDP.

             There are a number of INDEX objects that cannot be
             represented in the form of OBJECT clauses in SMIv2, but
             for which we have the following compliance
             requirements, expressed in OBJECT clause form in this
             description clause:

             -- OBJECT       udpEndpointLocalAddressType
             -- SYNTAX       InetAddressType { unknown(0), ipv4(1),
             --                                ipv6(2), ipv4z(3),
             --                                ipv6z(4) }
             -- DESCRIPTION
             --      Support for dns(5) is not required.
             -- OBJECT       udpEndpointLocalAddress
```

```
            -- SYNTAX          InetAddress (SIZE(0|4|8|16|20))
            -- DESCRIPTION
            --      Support is only required for zero-length
            --      octet-strings, and for scoped and unscoped
            --      IPv4 and IPv6 addresses.
            -- OBJECT          udpEndpointRemoteAddressType
            -- SYNTAX          InetAddressType { unknown(0), ipv4(1),
            --                                   ipv6(2), ipv4z(3),
            --                                   ipv6z(4) }
            -- DESCRIPTION
            --      Support for dns(5) is not required.
            -- OBJECT          udpEndpointRemoteAddress
            -- SYNTAX          InetAddress (SIZE(0|4|8|16|20))
            -- DESCRIPTION
            --      Support is only required for zero-length
            --      octet-strings, and for scoped and unscoped
            --      IPv4 and IPv6 addresses.
          "
MODULE   -- this module
        MANDATORY-GROUPS { udpBaseGroup, udpEndpointGroup }
        GROUP           udpHCGroup
        DESCRIPTION
                "This group is mandatory for systems that
                 are capable of receiving or transmitting more than
```

```
                    1 million UDP datagrams per second.  1 million
                    datagrams per second will cause a Counter32 to
                    wrap in just over an hour."
    ::= { udpMIBCompliances 2 }

udpMIBCompliance MODULE-COMPLIANCE
    STATUS      deprecated
    DESCRIPTION
            "The compliance statement for IPv4-only systems that
             implement UDP.  For IP version independence, this
             compliance statement is deprecated in favor of
             udpMIBCompliance2.  However, agents are still
             encouraged to implement these objects in order to
             interoperate with the deployed base of managers."
    MODULE  -- this module
        MANDATORY-GROUPS { udpGroup }
    ::= { udpMIBCompliances 1 }

-- units of conformance

udpGroup OBJECT-GROUP
    OBJECTS   { udpInDatagrams, udpNoPorts,
                udpInErrors, udpOutDatagrams,
                udpLocalAddress, udpLocalPort }
```

```
        STATUS        deprecated
        DESCRIPTION
                "The deprecated group of objects providing for
                 management of UDP over IPv4."
        ::= { udpMIBGroups 1 }

udpBaseGroup OBJECT-GROUP
        OBJECTS     { udpInDatagrams, udpNoPorts, udpInErrors,
                      udpOutDatagrams }
        STATUS        current
        DESCRIPTION
                "The group of objects providing for counters of UDP
                 statistics."
        ::= { udpMIBGroups 2 }

udpHCGroup OBJECT-GROUP
        OBJECTS     { udpHCInDatagrams, udpHCOutDatagrams }
        STATUS        current
        DESCRIPTION
                "The group of objects providing for counters of high
                 speed UDP implementations."
        ::= { udpMIBGroups 3 }

udpEndpointGroup OBJECT-GROUP
        OBJECTS       { udpEndpointProcess }
```

```
    STATUS        current
    DESCRIPTION
            "The group of objects providing for the IP version
             independent management of UDP 'endpoints'."
    ::= { udpMIBGroups 4 }

END
```