

Network Working Group  
Request for Comments: 2465  
Category: Standards Track

D. Haskin  
S. Onishi  
Bay Networks, Inc.  
December 1998

Management Information Base for IP Version 6:  
Textual Conventions and General Group

Status of this Memo

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

Copyright Notice

Copyright (C) The Internet Society (1998). All Rights Reserved.

Abstract

This document is one in the series of documents that provide MIB definitions for for IP Version 6. Specifically, the IPv6 MIB textual conventions as well as the IPv6 MIB General group is defined in this document.

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the IPv6-based internets.

This document specifies a MIB module in a manner that is both compliant to the SNMPv2 SMI, and semantically identical to the peer SNMPv1 definitions.

Table of Contents

1. The SNMPv2 Network Management Framework .....	2
1.1 Object Definitions .....	2
2. Overview .....	2
3. IPv6 Address Representation .....	3
4. Definition of Textual Conventions .....	4
5. The IPv6 General Group .....	5
6. Acknowledgments .....	36
7. References .....	36
8. Security Considerations .....	37
9. Authors' Addresses.....	37

## 10. Full Copyright Statement..... 38

## 1. The SNMPv2 Network Management Framework

The SNMPv2 Network Management Framework presently consists of three major components. They are:

- o the SMI, described in RFC 1902 [1] - the mechanisms used for describing and naming objects for the purpose of management.
- o the MIB-II, described in RFC 1213/STD 17 [3] - the core set of managed objects for the Internet suite of protocols.
- o RFC 1157/STD 15 [4] and RFC 1905 [5] which define two versions of the protocol used for network access to managed objects.

The Framework permits new objects to be defined for the purpose of experimentation and evaluation.

## 1.1. Object Definitions

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the subset of Abstract Syntax Notation One (ASN.1) defined in the SMI. In particular, each object type is named by an OBJECT IDENTIFIER, an administratively assigned name. The object type together with an object instance serves to uniquely identify a specific instantiation of the object. For human convenience, we often use a textual string, termed the descriptor, to refer to the object type.

## 2. Overview

This document is the first in the series of documents that define various MIB object groups for IPv6. These groups are the basic unit of conformance: if the semantics of a group is applicable to an implementation, then it must implement all objects in that group. For example, an implementation must implement the TCP group if and only if it implements the TCP over IPv6 protocol. At minimum, implementations must implement the IPv6 General group defined in this document as well as the ICMPv6 group [9].

This document defines the IPv6 MIB textual conventions as well as the IPv6 General group which provides for the basic management of IPv6 entities and serve as the foundation for other IPv6 MIB definitions.

The IPv6 General group consists of 6 tables:

- ipv6IfTable

The IPv6 Interfaces table contains information on the entity's IPv6 interfaces.

- ipv6IfStatsTable

This table contains information on the traffic statistics of the entity's IPv6 interfaces.

- ipv6AddrPrefixTable

The IPv6 Address Prefix table contains information on Address Prefixes that are associated with the entity's IPv6 interfaces.

- ipv6AddrTable

This table contains the addressing information relevant to the entity's IPv6 interfaces.

- ipv6RouteTable

The IPv6 routing table contains an entry for each valid IPv6 unicast route that can be used for packet forwarding determination.

- ipv6NetToMediaTable

The IPv6 address translation table contain the IPv6 Address to 'physical' address equivalencies.

### 3. IPv6 Address Representation

The IPv6 MIB defined in this memo uses an OCTET STRING of length 16 to represent 128-bit IPv6 address in network byte- order. This approach allows to implement IPv6 MIB without requiring any changes to the SNMPv2 SMI and compliant SNMP implementations.

## 4. Definition of Textual Conventions

```
IPV6-TC DEFINITIONS ::= BEGIN

IMPORTS
    Integer32                FROM SNMPv2-SMI
    TEXTUAL-CONVENTION       FROM SNMPv2-TC;

-- definition of textual conventions
Ipv6Address ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "2x:"
    STATUS      current
    DESCRIPTION
        "This data type is used to model IPv6 addresses.
         This is a binary string of 16 octets in network
         byte-order."
    SYNTAX      OCTET STRING (SIZE (16))

Ipv6AddressPrefix ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "2x:"
    STATUS      current
    DESCRIPTION
        "This data type is used to model IPv6 address
         prefixes. This is a binary string of up to 16
         octets in network byte-order."
    SYNTAX      OCTET STRING (SIZE (0..16))

Ipv6AddressIfIdentifier ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "2x:"
    STATUS      current
    DESCRIPTION
        "This data type is used to model IPv6 address
         interface identifiers. This is a binary string
         of up to 8 octets in network byte-order."
    SYNTAX      OCTET STRING (SIZE (0..8))

Ipv6IfIndex ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS      current
    DESCRIPTION
        "A unique value, greater than zero for each
         internetwork-layer interface in the managed
         system. It is recommended that values are assigned
         contiguously starting from 1. The value for each
         internetwork-layer interface must remain constant
         at least from one re-initialization of the entity's
         network management system to the next"
```

```

        re-initialization."
SYNTAX      Integer32 (1..2147483647)

Ipv6IfIndexOrZero ::= TEXTUAL-CONVENTION
DISPLAY-HINT "d"
STATUS      current
DESCRIPTION
    "This textual convention is an extension of the
    Ipv6IfIndex convention. The latter defines
    a greater than zero value used to identify an IPv6
    interface in the managed system. This extension
    permits the additional value of zero. The value
    zero is object-specific and must therefore be
    defined as part of the description of any object
    which uses this syntax. Examples of the usage of
    zero might include situations where interface was
    unknown, or when none or all interfaces need to be
    referenced."
SYNTAX      Integer32 (0..2147483647)

END

```

## 5. The IPv6 General Group

```

IPV6-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
    mib-2, Counter32, Unsigned32, Integer32,
    Gauge32                                FROM SNMPv2-SMI
    DisplayString, PhysAddress, TruthValue, TimeStamp,
    VariablePointer, RowPointer            FROM SNMPv2-TC
    MODULE-COMPLIANCE, OBJECT-GROUP,
    NOTIFICATION-GROUP                    FROM SNMPv2-CONF
    Ipv6IfIndex, Ipv6Address, Ipv6AddressPrefix,
    Ipv6AddressIfIdentifier,
    Ipv6IfIndexOrZero                     FROM IPV6-TC;

ipv6MIB MODULE-IDENTITY
    LAST-UPDATED "9802052155Z"
    ORGANIZATION "IETF IPv6 Working Group"
    CONTACT-INFO
        "
            Dimitry Haskin

            Postal: Bay Networks, Inc.
                   660 Techology Park Drive.
                   Billerica, MA 01821

```

US

Tel: +1-978-916-8124  
E-mail: dhaskin@baynetworks.com

Steve Onishi

Postal: Bay Networks, Inc.  
3 Federal Street  
Billerica, MA 01821  
US

Tel: +1-978-916-3816  
E-mail: sonishi@baynetworks.com"

DESCRIPTION

"The MIB module for entities implementing the IPv6 protocol."

::= { mib-2 55 }

-- the IPv6 general group

ipv6MIBObjects OBJECT IDENTIFIER ::= { ipv6MIB 1 }

ipv6Forwarding OBJECT-TYPE

SYNTAX INTEGER {  
forwarding(1), -- acting as a router  
  
notForwarding(2) -- NOT acting as  
a router  
}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The indication of whether this entity is acting as an IPv6 router in respect to the forwarding of datagrams received by, but not addressed to, this entity. IPv6 routers forward datagrams. IPv6 hosts do not (except those source-routed via the host).

Note that for some managed nodes, this object may take on only a subset of the values possible. Accordingly, it is appropriate for an agent to return a 'wrongValue' response if a management station attempts to change this object to an inappropriate value."

```
 ::= { ipv6MIBObjects 1 }
```

```
ipv6DefaultHopLimit OBJECT-TYPE
```

```
SYNTAX      INTEGER(0..255)
```

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

```
STATUS      current
```

```
DESCRIPTION
```

```
    "The default value inserted into the Hop Limit
     field of the IPv6 header of datagrams originated
     at this entity, whenever a Hop Limit value is not
     supplied by the transport layer protocol."
```

```
DEFVAL { 64 }
```

```
 ::= { ipv6MIBObjects 2 }
```

```
ipv6Interfaces OBJECT-TYPE
```

```
SYNTAX      Unsigned32
```

```
MAX-ACCESS  read-only
```

```
STATUS      current
```

```
DESCRIPTION
```

```
    "The number of IPv6 interfaces (regardless of
     their current state) present on this system."
```

```
 ::= { ipv6MIBObjects 3 }
```

```
ipv6IfTableLastChange OBJECT-TYPE
```

```
SYNTAX      TimeStamp
```

```
MAX-ACCESS  read-only
```

```
STATUS      current
```

```
DESCRIPTION
```

```
    "The value of sysUpTime at the time of the last
     insertion or removal of an entry in the
     ipv6IfTable. If the number of entries has been
     unchanged since the last re-initialization of
     the local network management subsystem, then this
     object contains a zero value."
```

```
 ::= { ipv6MIBObjects 4 }
```

```
-- the IPv6 Interfaces table
```

```
ipv6IfTable OBJECT-TYPE
```

```
SYNTAX      SEQUENCE OF Ipv6IfEntry
```

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

```
STATUS      current
```

```
DESCRIPTION
```

```
    "The IPv6 Interfaces table contains information
     on the entity's internetwork-layer interfaces.
     An IPv6 interface constitutes a logical network
     layer attachment to the layer immediately below
```

IPv6 including internet layer 'tunnels', such as tunnels over IPv4 or IPv6 itself."  
 ::= { ipv6MIBObjects 5 }

```

ipv6IfEntry OBJECT-TYPE
    SYNTAX      Ipv6IfEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An interface entry containing objects
         about a particular IPv6 interface."
    INDEX      { ipv6IfIndex }
    ::= { ipv6IfTable 1 }

Ipv6IfEntry ::= SEQUENCE {
    ipv6IfIndex          Ipv6IfIndex,
    ipv6IfDescr          DisplayString,
    ipv6IfLowerLayer     VariablePointer,
    ipv6IfEffectiveMtu   Unsigned32,
    ipv6IfReasmMaxSize   Unsigned32,
    ipv6IfIdentifier     Ipv6AddressIfIdentifier,
    ipv6IfIdentifierLength INTEGER,
    ipv6IfPhysicalAddress PhysAddress,
    ipv6IfAdminStatus    INTEGER,
    ipv6IfOperStatus     INTEGER,
    ipv6IfLastChange     TimeStamp
}

ipv6IfIndex OBJECT-TYPE
    SYNTAX      Ipv6IfIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A unique non-zero value identifying
         the particular IPv6 interface."
    ::= { ipv6IfEntry 1 }

ipv6IfDescr OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "A textual string containing information about the
         interface. This string may be set by the network
         management system."
    ::= { ipv6IfEntry 2 }

ipv6IfLowerLayer OBJECT-TYPE

```

SYNTAX VariablePointer  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION

"This object identifies the protocol layer over which this network interface operates. If this network interface operates over the data-link layer, then the value of this object refers to an instance of ifIndex [6]. If this network interface operates over an IPv4 interface, the value of this object refers to an instance of ipAdEntAddr [3].

If this network interface operates over another IPv6 interface, the value of this object refers to an instance of ipv6IfIndex. If this network interface is not currently operating over an active protocol layer, then the value of this object should be set to the OBJECT ID { 0 0 }."

::= { ipv6IfEntry 3 }

ipv6IfEffectiveMtu OBJECT-TYPE

SYNTAX Unsigned32  
 UNITS "octets"  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION

"The size of the largest IPv6 packet which can be sent/received on the interface, specified in octets."

::= { ipv6IfEntry 4 }

ipv6IfReasmMaxSize OBJECT-TYPE

SYNTAX Unsigned32 (0..65535)  
 UNITS "octets"  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION

"The size of the largest IPv6 datagram which this entity can re-assemble from incoming IPv6 fragmented datagrams received on this interface."

::= { ipv6IfEntry 5 }

ipv6IfIdentifier OBJECT-TYPE

SYNTAX Ipv6AddressIfIdentifier  
 MAX-ACCESS read-write  
 STATUS current  
 DESCRIPTION

"The Interface Identifier for this interface that

is (at least) unique on the link this interface is attached to. The Interface Identifier is combined with an address prefix to form an interface address.

By default, the Interface Identifier is autoconfigured according to the rules of the link type this interface is attached to."

```
::= { ipv6IfEntry 6 }
```

```
ipv6IfIdentifierLength OBJECT-TYPE
```

```
SYNTAX      INTEGER (0..64)
```

```
UNITS       "bits"
```

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

```
STATUS      current
```

```
DESCRIPTION
```

```
"The length of the Interface Identifier in bits."
```

```
::= { ipv6IfEntry 7 }
```

```
ipv6IfPhysicalAddress OBJECT-TYPE
```

```
SYNTAX      PhysAddress
```

```
MAX-ACCESS  read-only
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"The interface's physical address. For example, for an IPv6 interface attached to an 802.x link, this object normally contains a MAC address. Note that in some cases this address may differ from the address of the interface's protocol sub-layer. The interface's media-specific MIB must define the bit and byte ordering and the format of the value of this object. For interfaces which do not have such an address (e.g., a serial line), this object should contain an octet string of zero length."
```

```
::= { ipv6IfEntry 8 }
```

```
ipv6IfAdminStatus OBJECT-TYPE
```

```
SYNTAX      INTEGER {
                up(1),          -- ready to pass packets
                down(2)
            }
```

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

```
STATUS      current
```

```
DESCRIPTION
```

```
"The desired state of the interface. When a managed system initializes, all IPv6 interfaces start with ipv6IfAdminStatus in the down(2) state. As a result of either explicit management action or per configuration information retained by the managed
```

system, ipv6IfAdminStatus is then changed to the up(1) state (or remains in the down(2) state)."  
 ::= { ipv6IfEntry 9 }

ipv6IfOperStatus OBJECT-TYPE

```
SYNTAX  INTEGER {
    up(1),                -- ready to pass packets

    down(2),

    noIfIdentifier(3), -- no interface identifier

                                -- status can not be
                                -- determined for some
    unknown(4),          -- reason

                                -- some component is
                                -- missing
    notPresent(5)
}
```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The current operational state of the interface. The noIfIdentifier(3) state indicates that no valid Interface Identifier is assigned to the interface. This state usually indicates that the link-local interface address failed Duplicate Address Detection. If ipv6IfAdminStatus is down(2) then ipv6IfOperStatus should be down(2). If ipv6IfAdminStatus is changed to up(1) then ipv6IfOperStatus should change to up(1) if the interface is ready to transmit and receive network traffic; it should remain in the down(2) or noIfIdentifier(3) state if and only if there is a fault that prevents it from going to the up(1) state; it should remain in the notPresent(5) state if the interface has missing (typically, lower layer) components."

::= { ipv6IfEntry 10 }

ipv6IfLastChange OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of sysUpTime at the time the interface entered its current operational state. If the current state was entered prior to the last re-initialization of the local network management

```
        subsystem, then this object contains a zero
        value."
 ::= { ipv6IfEntry 11 }

-- IPv6 Interface Statistics table

ipv6IfStatsTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ipv6IfStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "IPv6 interface traffic statistics."
 ::= { ipv6MIBObjects 6 }

ipv6IfStatsEntry OBJECT-TYPE
    SYNTAX      Ipv6IfStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An interface statistics entry containing objects
        at a particular IPv6 interface."
    AUGMENTS { ipv6IfEntry }
 ::= { ipv6IfStatsTable 1 }

Ipv6IfStatsEntry ::= SEQUENCE {
    ipv6IfStatsInReceives
        Counter32,
    ipv6IfStatsInHdrErrors
        Counter32,
    ipv6IfStatsInTooBigErrors
        Counter32,
    ipv6IfStatsInNoRoutes
        Counter32,
    ipv6IfStatsInAddrErrors
        Counter32,
    ipv6IfStatsInUnknownProtos
        Counter32,
    ipv6IfStatsInTruncatedPkts
        Counter32,
    ipv6IfStatsInDiscards
        Counter32,
    ipv6IfStatsInDelivers
        Counter32,
    ipv6IfStatsOutForwDatagrams
        Counter32,
    ipv6IfStatsOutRequests
        Counter32,
    ipv6IfStatsOutDiscards
```

```

        Counter32,
    ipv6IfStatsOutFragOKs
        Counter32,
    ipv6IfStatsOutFragFails
        Counter32,
    ipv6IfStatsOutFragCreates
        Counter32,
    ipv6IfStatsReasmReqds
        Counter32,
    ipv6IfStatsReasmOKs
        Counter32,
    ipv6IfStatsReasmFails
        Counter32,
    ipv6IfStatsInMcastPkts
        Counter32,
    ipv6IfStatsOutMcastPkts
        Counter32
}

ipv6IfStatsInReceives OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The total number of input datagrams received by
         the interface, including those received in error."
    ::= { ipv6IfStatsEntry 1 }

ipv6IfStatsInHdrErrors OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of input datagrams discarded due to
         errors in their IPv6 headers, including version
         number mismatch, other format errors, hop count
         exceeded, errors discovered in processing their
         IPv6 options, etc."
    ::= { ipv6IfStatsEntry 2 }

ipv6IfStatsInTooBigErrors OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of input datagrams that could not be
         forwarded because their size exceeded the link MTU
         of outgoing interface."

```

```
::= { ipv6IfStatsEntry 3 }
```

```
ipv6IfStatsInNoRoutes OBJECT-TYPE
```

```
SYNTAX Counter32
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"The number of input datagrams discarded because no route could be found to transmit them to their destination."
```

```
::= { ipv6IfStatsEntry 4 }
```

```
ipv6IfStatsInAddrErrors OBJECT-TYPE
```

```
SYNTAX Counter32
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"The number of input datagrams discarded because the IPv6 address in their IPv6 header's destination field was not a valid address to be received at this entity. This count includes invalid addresses (e.g., ::0) and unsupported addresses (e.g., addresses with unallocated prefixes). For entities which are not IPv6 routers and therefore do not forward datagrams, this counter includes datagrams discarded because the destination address was not a local address."
```

```
::= { ipv6IfStatsEntry 5 }
```

```
ipv6IfStatsInUnknownProtos OBJECT-TYPE
```

```
SYNTAX Counter32
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"The number of locally-addressed datagrams received successfully but discarded because of an unknown or unsupported protocol. This counter is incremented at the interface to which these datagrams were addressed which might not be necessarily the input interface for some of the datagrams."
```

```
::= { ipv6IfStatsEntry 6 }
```

```
ipv6IfStatsInTruncatedPkts OBJECT-TYPE
```

```
SYNTAX Counter32
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

## DESCRIPTION

"The number of input datagrams discarded because datagram frame didn't carry enough data."

::= { ipv6IfStatsEntry 7 }

## ipv6IfStatsInDiscards OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The number of input IPv6 datagrams for which no problems were encountered to prevent their continued processing, but which were discarded (e.g., for lack of buffer space). Note that this counter does not include any datagrams discarded while awaiting re-assembly."

::= { ipv6IfStatsEntry 8 }

## ipv6IfStatsInDelivers OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The total number of datagrams successfully delivered to IPv6 user-protocols (including ICMP). This counter is incremented at the interface to which these datagrams were addressed which might not be necessarily the input interface for some of the datagrams."

::= { ipv6IfStatsEntry 9 }

## ipv6IfStatsOutForwDatagrams OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The number of output datagrams which this entity received and forwarded to their final destinations. In entities which do not act as IPv6 routers, this counter will include only those packets which were Source-Routed via this entity, and the Source-Route processing was successful. Note that for a successfully forwarded datagram the counter of the outgoing interface is incremented."

::= { ipv6IfStatsEntry 10 }

## ipv6IfStatsOutRequests OBJECT-TYPE

SYNTAX Counter32  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
"The total number of IPv6 datagrams which local IPv6 user-protocols (including ICMP) supplied to IPv6 in requests for transmission. Note that this counter does not include any datagrams counted in ipv6IfStatsOutForwDatagrams."  
 ::= { ipv6IfStatsEntry 11 }

ipv6IfStatsOutDiscards OBJECT-TYPE

SYNTAX Counter32  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
"The number of output IPv6 datagrams for which no problem was encountered to prevent their transmission to their destination, but which were discarded (e.g., for lack of buffer space). Note that this counter would include datagrams counted in ipv6IfStatsOutForwDatagrams if any such packets met this (discretionary) discard criterion."  
 ::= { ipv6IfStatsEntry 12 }

ipv6IfStatsOutFragOKs OBJECT-TYPE

SYNTAX Counter32  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
"The number of IPv6 datagrams that have been successfully fragmented at this output interface."  
 ::= { ipv6IfStatsEntry 13 }

ipv6IfStatsOutFragFails OBJECT-TYPE

SYNTAX Counter32  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
"The number of IPv6 datagrams that have been discarded because they needed to be fragmented at this output interface but could not be."  
 ::= { ipv6IfStatsEntry 14 }

ipv6IfStatsOutFragCreates OBJECT-TYPE

SYNTAX Counter32  
MAX-ACCESS read-only  
STATUS current

## DESCRIPTION

"The number of output datagram fragments that have been generated as a result of fragmentation at this output interface."

::= { ipv6IfStatsEntry 15 }

## ipv6IfStatsReasmReqds OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The number of IPv6 fragments received which needed to be reassembled at this interface. Note that this counter is incremented at the interface to which these fragments were addressed which might not be necessarily the input interface for some of the fragments."

::= { ipv6IfStatsEntry 16 }

## ipv6IfStatsReasmOKs OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The number of IPv6 datagrams successfully reassembled. Note that this counter is incremented at the interface to which these datagrams were addressed which might not be necessarily the input interface for some of the fragments."

::= { ipv6IfStatsEntry 17 }

## ipv6IfStatsReasmFails OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The number of failures detected by the IPv6 re-assembly algorithm (for whatever reason: timed out, errors, etc.). Note that this is not necessarily a count of discarded IPv6 fragments since some algorithms (notably the algorithm in RFC 815) can lose track of the number of fragments by combining them as they are received. This counter is incremented at the interface to which these fragments were addressed which might not be necessarily the input interface for some of the fragments."

::= { ipv6IfStatsEntry 18 }

```

ipv6IfStatsInMcastPkts OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of multicast packets received
         by the interface"
    ::= { ipv6IfStatsEntry 19 }

ipv6IfStatsOutMcastPkts OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of multicast packets transmitted
         by the interface"
    ::= { ipv6IfStatsEntry 20 }

-- Address Prefix table

-- The IPv6 Address Prefix table contains information on
-- the entity's IPv6 Address Prefixes that are associated
-- with IPv6 interfaces.

ipv6AddrPrefixTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ipv6AddrPrefixEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The list of IPv6 address prefixes of
         IPv6 interfaces."
    ::= { ipv6MIBObjects 7 }

ipv6AddrPrefixEntry OBJECT-TYPE
    SYNTAX      Ipv6AddrPrefixEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An interface entry containing objects of
         a particular IPv6 address prefix."
    INDEX      { ipv6IfIndex,
                 ipv6AddrPrefix,
                 ipv6AddrPrefixLength }
    ::= { ipv6AddrPrefixTable 1 }

Ipv6AddrPrefixEntry ::= SEQUENCE {

```

```

        ipv6AddrPrefix                Ipv6AddressPrefix,
        ipv6AddrPrefixLength          INTEGER (0..128),
        ipv6AddrPrefixOnLinkFlag      TruthValue,
        ipv6AddrPrefixAutonomousFlag  TruthValue,
        ipv6AddrPrefixAdvPreferredLifetime Unsigned32,
        ipv6AddrPrefixAdvValidLifetime Unsigned32
    }

```

ipv6AddrPrefix OBJECT-TYPE

```

SYNTAX      Ipv6AddressPrefix
MAX-ACCESS  not-accessible
STATUS      current

```

DESCRIPTION

"The prefix associated with the this interface."

```
 ::= { ipv6AddrPrefixEntry 1 }
```

ipv6AddrPrefixLength OBJECT-TYPE

```

SYNTAX      INTEGER (0..128)
UNITS       "bits"
MAX-ACCESS  not-accessible
STATUS      current

```

DESCRIPTION

"The length of the prefix (in bits)."

```
 ::= { ipv6AddrPrefixEntry 2 }
```

ipv6AddrPrefixOnLinkFlag OBJECT-TYPE

```

SYNTAX      TruthValue
MAX-ACCESS  read-only
STATUS      current

```

DESCRIPTION

"This object has the value 'true(1)', if this prefix can be used for on-link determination and the value 'false(2)' otherwise."

```
 ::= { ipv6AddrPrefixEntry 3 }
```

ipv6AddrPrefixAutonomousFlag OBJECT-TYPE

```

SYNTAX      TruthValue
MAX-ACCESS  read-only
STATUS      current

```

DESCRIPTION

"Autonomous address configuration flag. When true(1), indicates that this prefix can be used for autonomous address configuration (i.e. can be used to form a local interface address). If false(2), it is not used to autoconfigure a local interface address."

```
 ::= { ipv6AddrPrefixEntry 4 }
```

```

ipv6AddrPrefixAdvPreferredLifetime OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "seconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "It is the length of time in seconds that this
        prefix will remain preferred, i.e. time until
        deprecation.  A value of 4,294,967,295 represents
        infinity.

        The address generated from a deprecated prefix
        should no longer be used as a source address in
        new communications, but packets received on such
        an interface are processed as expected."
    ::= { ipv6AddrPrefixEntry 5 }

```

```

ipv6AddrPrefixAdvValidLifetime OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "seconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "It is the length of time in seconds that this
        prefix will remain valid, i.e. time until
        invalidation.  A value of 4,294,967,295 represents
        infinity.

        The address generated from an invalidated prefix
        should not appear as the destination or source
        address of a packet."
    ::= { ipv6AddrPrefixEntry 6 }

```

-- the IPv6 Address table

-- The IPv6 address table contains this node's IPv6  
 -- addressing information.

```

ipv6AddrTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ipv6AddrEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The table of addressing information relevant to
        this node's interface addresses."
    ::= { ipv6MIBObjects 8 }

```

```

ipv6AddrEntry OBJECT-TYPE
    SYNTAX      Ipv6AddrEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The addressing information for one of this
         node's interface addresses."
    INDEX      { ipv6IfIndex, ipv6AddrAddress }
    ::= { ipv6AddrTable 1 }

Ipv6AddrEntry ::=
    SEQUENCE {
        ipv6AddrAddress      Ipv6Address,
        ipv6AddrPfxLength    INTEGER,
        ipv6AddrType         INTEGER,
        ipv6AddrAnycastFlag  TruthValue,
        ipv6AddrStatus       INTEGER
    }

ipv6AddrAddress OBJECT-TYPE
    SYNTAX      Ipv6Address
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The IPv6 address to which this entry's addressing
         information pertains."
    ::= { ipv6AddrEntry 1 }

ipv6AddrPfxLength OBJECT-TYPE
    SYNTAX      INTEGER(0..128)
    UNITS       "bits"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The length of the prefix (in bits) associated with
         the IPv6 address of this entry."
    ::= { ipv6AddrEntry 2 }

ipv6AddrType OBJECT-TYPE
    SYNTAX      INTEGER {
        -- address has been formed
        -- using stateless
        stateless(1), -- autoconfiguration

        -- address has been acquired
        -- by stateful means
        -- (e.g. DHCPv6, manual
        stateful(2),  -- configuration)
    }

```

```

        unknown(3)      -- type can not be determined
                        -- for some reason.
    }
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION   "The type of address. Note that 'stateless(1)'
               refers to an address that was statelessly
               autoconfigured; 'stateful(2)' refers to a address
               which was acquired by via a stateful protocol
               (e.g. DHCPv6, manual configuration)."
 ::= { ipv6AddrEntry 3 }

ipv6AddrAnycastFlag OBJECT-TYPE
SYNTAX        TruthValue
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION   "This object has the value 'true(1)', if this
               address is an anycast address and the value
               'false(2)' otherwise."
 ::= { ipv6AddrEntry 4 }

ipv6AddrStatus OBJECT-TYPE
SYNTAX        INTEGER {
               preferred(1),

               deprecated(2),

               invalid(3),

               inaccessible(4),

               unknown(5)  -- status can not be determined
                           -- for some reason.
               }
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION   "Address status. The preferred(1) state indicates
               that this is a valid address that can appear as
               the destination or source address of a packet.
               The deprecated(2) state indicates that this is
               a valid but deprecated address that should no longer
               be used as a source address in new communications,
               but packets addressed to such an address are
               processed as expected. The invalid(3) state indicates
               that this is not valid address which should not

```

appear as the destination or source address of a packet. The inaccessible(4) state indicates that the address is not accessible because the interface to which this address is assigned is not operational."  
 ::= { ipv6AddrEntry 5 }

-- IPv6 Routing objects

ipv6RouteNumber OBJECT-TYPE  
SYNTAX Gauge32  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
"The number of current ipv6RouteTable entries. This is primarily to avoid having to read the table in order to determine this number."  
 ::= { ipv6MIBObjects 9 }

ipv6DiscardedRoutes OBJECT-TYPE  
SYNTAX Counter32  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
"The number of routing entries which were chosen to be discarded even though they are valid. One possible reason for discarding such an entry could be to free-up buffer space for other routing entries."  
 ::= { ipv6MIBObjects 10 }

-- IPv6 Routing table

ipv6RouteTable OBJECT-TYPE  
SYNTAX SEQUENCE OF Ipv6RouteEntry  
MAX-ACCESS not-accessible  
STATUS current  
DESCRIPTION  
"IPv6 Routing table. This table contains an entry for each valid IPv6 unicast route that can be used for packet forwarding determination."  
 ::= { ipv6MIBObjects 11 }

ipv6RouteEntry OBJECT-TYPE  
SYNTAX Ipv6RouteEntry  
MAX-ACCESS not-accessible

```

STATUS      current
DESCRIPTION
    "A routing entry."
INDEX       { ipv6RouteDest,
              ipv6RoutePfxLength,
              ipv6RouteIndex }
 ::= { ipv6RouteTable 1 }

```

```

Ipv6RouteEntry ::= SEQUENCE {
    ipv6RouteDest      Ipv6Address,
    ipv6RoutePfxLength INTEGER,
    ipv6RouteIndex     Unsigned32,
    ipv6RouteIfIndex   Ipv6IfIndexOrZero,
    ipv6RouteNextHop   Ipv6Address,
    ipv6RouteType       INTEGER,
    ipv6RouteProtocol  INTEGER,
    ipv6RoutePolicy     Integer32,
    ipv6RouteAge        Unsigned32,
    ipv6RouteNextHopRDI Unsigned32,
    ipv6RouteMetric     Unsigned32,
    ipv6RouteWeight     Unsigned32,
    ipv6RouteInfo       RowPointer,
    ipv6RouteValid      TruthValue
}

```

```

ipv6RouteDest OBJECT-TYPE
SYNTAX      Ipv6Address
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "The destination IPv6 address of this route.
    This object may not take a Multicast address
    value."
 ::= { ipv6RouteEntry 1 }

```

```

ipv6RoutePfxLength OBJECT-TYPE
SYNTAX      INTEGER(0..128)
UNITS       "bits"
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "Indicates the prefix length of the destination
    address."
 ::= { ipv6RouteEntry 2 }

```

```

ipv6RouteIndex OBJECT-TYPE
SYNTAX      Unsigned32
MAX-ACCESS  not-accessible

```

```

STATUS      current
DESCRIPTION
  "The value which uniquely identifies the route
  among the routes to the same network layer
  destination.  The way this value is chosen is
  implementation specific but it must be unique for
  ipv6RouteDest/ipv6RoutePfxLength pair and remain
  constant for the life of the route."
 ::= { ipv6RouteEntry 3 }

```

```

ipv6RouteIfIndex OBJECT-TYPE
SYNTAX      Ipv6IfIndexOrZero
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
  "The index value which uniquely identifies the local
  interface through which the next hop of this
  route should be reached.  The interface identified
  by a particular value of this index is the same
  interface as identified by the same value of
  ipv6IfIndex.  For routes of the discard type this
  value can be zero."
 ::= { ipv6RouteEntry 4 }

```

```

ipv6RouteNextHop OBJECT-TYPE
SYNTAX      Ipv6Address
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
  "On remote routes, the address of the next
  system en route; otherwise, ::0
  ('00000000000000000000000000000000'H in ASN.1
  string representation)."
 ::= { ipv6RouteEntry 5 }

```

```

ipv6RouteType OBJECT-TYPE
SYNTAX      INTEGER {
  other(1),      -- none of the following
                 -- an route indicating that
                 -- packets to destinations
                 -- matching this route are
  discard(2),   -- to be discarded
                 -- route to directly
  local(3),     -- connected (sub-)network
                 -- route to a remote

```

```

        remote(4)      -- destination
    }
    MAX-ACCESS read-only
    STATUS      current
    DESCRIPTION
        "The type of route. Note that 'local(3)' refers
        to a route for which the next hop is the final
        destination; 'remote(4)' refers to a route for
        which the next hop is not the final
        destination; 'discard(2)' refers to a route
        indicating that packets to destinations matching
        this route are to be discarded (sometimes called
        black-hole route)."
    ::= { ipv6RouteEntry 6 }

ipv6RouteProtocol OBJECT-TYPE
    SYNTAX      INTEGER {
        other(1),      -- none of the following
                        -- non-protocol information,
                        -- e.g., manually configured
        local(2),     -- entries
        netmgmt(3),  -- static route
                        -- obtained via Neighbor
                        -- Discovery protocol,
        ndisc(4),    -- e.g., result of Redirect
                        -- the following are all
                        -- dynamic routing protocols
        rip(5),      -- RIPng
        ospf(6),     -- Open Shortest Path First
        bgp(7),      -- Border Gateway Protocol
        idrp(8),     -- InterDomain Routing Protocol
        igrp(9)      -- InterGateway Routing Protocol
    }
    MAX-ACCESS read-only
    STATUS      current
    DESCRIPTION
        "The routing mechanism via which this route was
        learned."
    ::= { ipv6RouteEntry 7 }

ipv6RoutePolicy OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS read-only

```

STATUS current

DESCRIPTION

"The general set of conditions that would cause the selection of one multipath route (set of next hops for a given destination) is referred to as 'policy'. Unless the mechanism indicated by ipv6RouteProtocol specified otherwise, the policy specifier is the 8-bit Traffic Class field of the IPv6 packet header that is zero extended at the left to a 32-bit value.

Protocols defining 'policy' otherwise must either define a set of values which are valid for this object or must implement an integer-instanced policy table for which this object's value acts as an index."

::= { ipv6RouteEntry 8 }

ipv6RouteAge OBJECT-TYPE

SYNTAX Unsigned32

UNITS "seconds"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of seconds since this route was last updated or otherwise determined to be correct.

Note that no semantics of 'too old' can be implied except through knowledge of the routing protocol by which the route was learned."

::= { ipv6RouteEntry 9 }

ipv6RouteNextHopRDI OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The Routing Domain ID of the Next Hop.

The semantics of this object are determined by the routing-protocol specified in the route's ipv6RouteProtocol value. When this object is unknown or not relevant its value should be set to zero."

::= { ipv6RouteEntry 10 }

ipv6RouteMetric OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The routing metric for this route. The semantics of this metric are determined by the routing protocol specified in the route's ipv6RouteProtocol value. When this is unknown or not relevant to the protocol indicated by ipv6RouteProtocol, the object value should be set to its maximum value (4,294,967,295)."

::= { ipv6RouteEntry 11 }

ipv6RouteWeight OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The system internal weight value for this route. The semantics of this value are determined by the implementation specific rules. Generally, within routes with the same ipv6RoutePolicy value, the lower the weight value the more preferred is the route."

::= { ipv6RouteEntry 12 }

ipv6RouteInfo OBJECT-TYPE

SYNTAX RowPointer

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"A reference to MIB definitions specific to the particular routing protocol which is responsible for this route, as determined by the value specified in the route's ipv6RouteProto value. If this information is not present, its value should be set to the OBJECT ID { 0 0 }, which is a syntactically valid object identifier, and any implementation conforming to ASN.1 and the Basic Encoding Rules must be able to generate and recognize this value."

::= { ipv6RouteEntry 13 }

ipv6RouteValid OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Setting this object to the value 'false(2)' has the effect of invalidating the corresponding entry in the ipv6RouteTable object. That is, it effectively disassociates the destination

identified with said entry from the route identified with said entry. It is an implementation-specific matter as to whether the agent removes an invalidated entry from the table. Accordingly, management stations must be prepared to receive tabular information from agents that corresponds to entries not currently in use. Proper interpretation of such entries requires examination of the relevant ipv6RouteValid object."

```
DEFVAL { true }
 ::= { ipv6RouteEntry 14 }
```

-- IPv6 Address Translation table

```
ipv6NetToMediaTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF Ipv6NetToMediaEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "The IPv6 Address Translation table used for
        mapping from IPv6 addresses to physical addresses.
```

The IPv6 address translation table contain the Ipv6Address to 'physical' address equivalencies. Some interfaces do not use translation tables for determining address equivalencies; if all interfaces are of this type, then the Address Translation table is empty, i.e., has zero entries."

```
 ::= { ipv6MIBObjects 12 }
```

```
ipv6NetToMediaEntry OBJECT-TYPE
    SYNTAX          Ipv6NetToMediaEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
```

"Each entry contains one IPv6 address to 'physical' address equivalence."

```
INDEX { ipv6IfIndex,
        ipv6NetToMediaNetAddress }
```

```
 ::= { ipv6NetToMediaTable 1 }
```

```
Ipv6NetToMediaEntry ::= SEQUENCE {
    ipv6NetToMediaNetAddress
        Ipv6Address,
    ipv6NetToMediaPhysAddress
```

```

        PhysAddress,
    ipv6NetToMediaType
        INTEGER,
    ipv6IfNetToMediaState
        INTEGER,
    ipv6IfNetToMediaLastUpdated
        TimeStamp,
    ipv6NetToMediaValid
        TruthValue
    }

    ipv6NetToMediaNetAddress OBJECT-TYPE
        SYNTAX      Ipv6Address
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "The IPv6 Address corresponding to
             the media-dependent 'physical' address."
        ::= { ipv6NetToMediaEntry 1 }

    ipv6NetToMediaPhysAddress OBJECT-TYPE
        SYNTAX      PhysAddress
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The media-dependent 'physical' address."
        ::= { ipv6NetToMediaEntry 2 }

    ipv6NetToMediaType OBJECT-TYPE
        SYNTAX      INTEGER {
            other(1),      -- none of the following
            dynamic(2),   -- dynamically resolved
            static(3),    -- statically configured
            local(4)      -- local interface
        }
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The type of the mapping. The 'dynamic(2)' type
             indicates that the IPv6 address to physical
             addresses mapping has been dynamically
             resolved using the IPv6 Neighbor Discovery
             protocol. The static(3)' types indicates that
             the mapping has been statically configured.
             The local(4) indicates that the mapping is
             provided for an entity's own interface address."
        ::= { ipv6NetToMediaEntry 3 }

```

```

ipv6IfNetToMediaState OBJECT-TYPE
    SYNTAX          INTEGER {
        reachable(1), -- confirmed reachability

        stale(2),    -- unconfirmed reachability

        delay(3),    -- waiting for reachability
                    -- confirmation before entering
                    -- the probe state

        probe(4),    -- actively probing

        invalid(5),  -- an invalidated mapping

        unknown(6)   -- state can not be determined
                    -- for some reason.
    }
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The Neighbor Unreachability Detection [8] state
        for the interface when the address mapping in
        this entry is used."
    ::= { ipv6NetToMediaEntry 4 }

ipv6IfNetToMediaLastUpdated OBJECT-TYPE
    SYNTAX          TimeStamp
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The value of sysUpTime at the time this entry
        was last updated.  If this entry was updated prior
        to the last re-initialization of the local network
        management subsystem, then this object contains
        a zero value."
    ::= { ipv6NetToMediaEntry 5 }

ipv6NetToMediaValid OBJECT-TYPE
    SYNTAX          TruthValue
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        "Setting this object to the value 'false(2)' has
        the effect of invalidating the corresponding entry
        in the ipv6NetToMediaTable.  That is, it effectively
        disassociates the interface identified with said
        entry from the mapping identified with said entry.
        It is an implementation-specific matter as to

```

whether the agent removes an invalidated entry from the table. Accordingly, management stations must be prepared to receive tabular information from agents that corresponds to entries not currently in use. Proper interpretation of such entries requires examination of the relevant ipv6NetToMediaValid object."

```
DEFVAL { true }
 ::= { ipv6NetToMediaEntry 6 }
```

```
-- definition of IPv6-related notifications.
-- Note that we need ipv6NotificationPrefix with the 0
-- sub-identifier to make this MIB to translate to
-- an SNMPv1 format in a reversible way. For example
-- it is needed for proxies that convert SNMPv1 traps
-- to SNMPv2 notifications without MIB knowledge.
```

```
ipv6Notifications      OBJECT IDENTIFIER
 ::= { ipv6MIB 2 }
```

```
ipv6NotificationPrefix OBJECT IDENTIFIER
 ::= { ipv6Notifications 0 }
```

```
ipv6IfStateChange NOTIFICATION-TYPE
 OBJECTS {
     ipv6IfDescr,
     ipv6IfOperStatus -- the new state of the If.
 }
```

```
STATUS current
```

```
DESCRIPTION
```

```
"An ipv6IfStateChange notification signifies
that there has been a change in the state of
an ipv6 interface. This notification should
be generated when the interface's operational
status transitions to or from the up(1) state."
```

```
 ::= { ipv6NotificationPrefix 1 }
```

```
-- conformance information
```

```
ipv6Conformance OBJECT IDENTIFIER ::= { ipv6MIB 3 }
```

```
ipv6Compliances OBJECT IDENTIFIER ::= { ipv6Conformance 1 }
```

```
ipv6Groups      OBJECT IDENTIFIER ::= { ipv6Conformance 2 }
```

```
-- compliance statements
```

```
ipv6Compliance MODULE-COMPLIANCE
  STATUS current
  DESCRIPTION
    "The compliance statement for SNMPv2 entities which
    implement ipv6 MIB."
  MODULE -- this module
    MANDATORY-GROUPS { ipv6GeneralGroup,
                       ipv6NotificationGroup }
    OBJECT ipv6Forwarding
      MIN-ACCESS read-only
      DESCRIPTION
        "An agent is not required to provide write
        access to this object"
    OBJECT ipv6DefaultHopLimit
      MIN-ACCESS read-only
      DESCRIPTION
        "An agent is not required to provide write
        access to this object"
    OBJECT ipv6IfDescr
      MIN-ACCESS read-only
      DESCRIPTION
        "An agent is not required to provide write
        access to this object"
    OBJECT ipv6IfIdentifier
      MIN-ACCESS read-only
      DESCRIPTION
        "An agent is not required to provide write
        access to this object"
    OBJECT ipv6IfIdentifierLength
      MIN-ACCESS read-only
      DESCRIPTION
        "An agent is not required to provide write
        access to this object"

    OBJECT ipv6IfAdminStatus
      MIN-ACCESS read-only
      DESCRIPTION
        "An agent is not required to provide write
        access to this object"
    OBJECT ipv6RouteValid
      MIN-ACCESS read-only
      DESCRIPTION
        "An agent is not required to provide write
        access to this object"
    OBJECT ipv6NetToMediaValid
      MIN-ACCESS read-only
      DESCRIPTION
        "An agent is not required to provide write
```

```
        access to this object"  
 ::= { ipv6Compliances 1 }
```

```
ipv6GeneralGroup OBJECT-GROUP  
  OBJECTS { ipv6Forwarding,  
            ipv6DefaultHopLimit,  
            ipv6Interfaces,  
            ipv6IfTableLastChange,  
            ipv6IfDescr,  
            ipv6IfLowerLayer,  
            ipv6IfEffectiveMtu,  
            ipv6IfReasmMaxSize,  
            ipv6IfIdentifier,  
            ipv6IfIdentifierLength,  
            ipv6IfPhysicalAddress,  
            ipv6IfAdminStatus,  
            ipv6IfOperStatus,  
            ipv6IfLastChange,  
            ipv6IfStatsInReceives,  
            ipv6IfStatsInHdrErrors,  
            ipv6IfStatsInTooBigErrors,  
            ipv6IfStatsInNoRoutes,  
            ipv6IfStatsInAddrErrors,  
            ipv6IfStatsInUnknownProtos,  
            ipv6IfStatsInTruncatedPkts,  
            ipv6IfStatsInDiscards,  
            ipv6IfStatsInDelivers,  
            ipv6IfStatsOutForwDatagrams,  
            ipv6IfStatsOutRequests,  
            ipv6IfStatsOutDiscards,  
            ipv6IfStatsOutFragOKs,  
            ipv6IfStatsOutFragFails,  
            ipv6IfStatsOutFragCreates,  
            ipv6IfStatsReasmReqds,  
            ipv6IfStatsReasmOKs,  
            ipv6IfStatsReasmFails,  
            ipv6IfStatsInMcastPkts,  
            ipv6IfStatsOutMcastPkts,  
            ipv6AddrPrefixOnLinkFlag,  
            ipv6AddrPrefixAutonomousFlag,  
            ipv6AddrPrefixAdvPreferredLifetime,  
            ipv6AddrPrefixAdvValidLifetime,  
            ipv6AddrPfxLength,  
            ipv6AddrType,  
            ipv6AddrAnycastFlag,  
            ipv6AddrStatus,  
            ipv6RouteNumber,  
            ipv6DiscardedRoutes,
```

```
        ipv6RouteIfIndex,
        ipv6RouteNextHop,
        ipv6RouteType,
        ipv6RouteProtocol,
        ipv6RoutePolicy,
        ipv6RouteAge,
        ipv6RouteNextHopRDI,
        ipv6RouteMetric,
        ipv6RouteWeight,
        ipv6RouteInfo,
        ipv6RouteValid,
        ipv6NetToMediaPhysAddress,
        ipv6NetToMediaType,
        ipv6IfNetToMediaState,
        ipv6IfNetToMediaLastUpdated,
        ipv6NetToMediaValid }
STATUS      current
DESCRIPTION
    "The IPv6 group of objects providing for basic
    management of IPv6 entities."
 ::= { ipv6Groups 1 }

ipv6NotificationGroup NOTIFICATION-GROUP
NOTIFICATIONS { ipv6IfStateChange }
STATUS      current
DESCRIPTION
    "The notification that an IPv6 entity is required
    to implement."

 ::= { ipv6Groups 2 }

END
```

## 6. Acknowledgments

This document borrows from MIB works produced by IETF for IPv4-based internets.

We would like to thanks the following individuals for constructive and valuable comments:

Mike Daniele,  
Margaret Forsythe,  
Tim Hartrick,  
Jean-Pierre Roch,  
Juergen Schoenwaelder,  
Frank Solensky,  
Vivek Venkatraman.

## 7. References

- [1] SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1902, January 1996.
- [2] SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1903, January 1996.
- [3] McCloghrie, K., and M. Rose, Editors, "Management Information Base for Network Management of TCP/IP-based internets: MIB-II", STD 17, RFC 1213, Hughes LAN Systems, Performance Systems International, March 1991.
- [4] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "A Simple Network Management Protocol (SNMP)", STD 15, RFC 1157, SNMP Research, Performance Systems International, MIT Lab for Computer Science, May 1990.
- [5] SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1905, January 1996.
- [6] McCloghrie, K. and F. Kastenholz, "Evolution of the Interfaces Group of MIB-II", RFC 1573, January 1994.
- [7] Deering, S., and R. Hinden, Editors, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

- [8] Narten, T., Nordmark E., and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, December 1998.
- [9] Haskin, D., and S. Onishi, "Management Information Base for IP Version 6: ICMPv6 Group", RFC 2466, December 1998.

## 8. Security Considerations

Certain management information defined in this MIB may be considered sensitive in some network environments.

Therefore, authentication of received SNMP requests and controlled access to management information should be employed in such environments.

## 9. Authors' Addresses

Dimitry Haskin  
Bay Networks, Inc.  
600 Technology Park Drive  
Billerica, MA 01821

EMail: [dhaskin@baynetworks.com](mailto:dhaskin@baynetworks.com)

Steve Onishi  
Bay Networks, Inc.  
3 Federal Street  
Billerica, MA 01821

EMail: [sonishi@baynetworks.com](mailto:sonishi@baynetworks.com)

## 10. Full Copyright Statement

Copyright (C) The Internet Society (1997). 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."

