

Network Working Group
Request for Comments: 4293
Obsoletes: 2011, 2465, 2466
Category: Standards Track

S. Routhier, Ed.
April 2006

Management Information Base
for the Internet Protocol (IP)

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 (2006).

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 describes managed objects used for implementations of the Internet Protocol (IP) in an IP version independent manner. This memo obsoletes RFCs 2011, 2465, and 2466.

Table of Contents

1. The Internet-Standard Management Framework	2
2. Revision History	3
3. Overview	3
3.1. Multi-Stack Implementations	3
3.2. Discussion of Tables and Groups	3
3.2.1. General Objects	4
3.2.2. Interface Tables	4
3.2.3. IP Statistics Tables	4
3.2.4. Internet Address Prefix Table	8
3.2.5. Internet Address Table	8
3.2.6. Internet Address Translation Table	9
3.2.7. IPv6 Scope Zone Index Table	9
3.2.8. Default Router Table	9
3.2.9. Router Advertisement Table	9
3.2.10. ICMP Statistics Tables	9
3.2.11. Conformance and Compliance	10
3.2.12. Deprecated Objects	10
4. Updating Implementations	10
4.1. Updating an Implementation of the IPv4-only IP-MIB	11
4.2. Updating an Implementation of the IPv6-MIB	12
5. Definitions	13
6. Previous Work	116
7. References	116
7.1. Normative References	116
7.2. Informative References	117
8. Security Considerations	118
9. Acknowledgements	120
10. Authors	120

1. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [9].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIV2, which is described in STD 58, RFC 2578 [1], STD 58, RFC 2579 [2] and STD 58, RFC 2580 [3].

2. Revision History

One of the primary purposes of this revision of the IP MIB is to create a single set of objects to describe and manage IP modules in an IP version independent manner. Where RFCs 2465 and 2466 created a set of objects independent from RFC 2011, this document merges those three documents into a single unified set of objects. The `ipSystemStatsTable` and `ipIfStatsTable` tables are examples of updating objects to be independent of IP version. Both of these tables contain counters to reflect IP traffic statistics that originated in much earlier MIBs and both include an IP address type in order to separate the information based on IP version.

Another purpose of this document is to increase the manageability of a node running IPv6 by adding new objects. Some of these tables, such as `ipDefaultRouterTable`, may be useful on both IPv4 and IPv6 nodes while others, such as `ipv6RouterAdvertTable`, are specific to a single protocol.

3. Overview

3.1. Multi-Stack Implementations

This MIB does not provide native support for implementations of multiple stacks sharing the same address type. One option for supporting such designs is to assign each stack within an address type to a separate context. These contexts could then be selected based upon the context name, with the Entity MIB and View-based Access Control Model (VACM) Context Table providing methods for listing the supported contexts.

3.2. Discussion of Tables and Groups

This MIB is composed of a small number of discrete objects and a series of tables meant to form the base for managing IPv4 and IPv6 entities.

While some of the objects are meant to be included in all entities, some of the objects are only conditionally mandatory. The unconditionally mandatory objects are mostly counters for IP and ICMP statistics. The conditionally mandatory objects fall into one of several groups: objects for use in higher bandwidth situations, objects for use with IPv4, objects for use with IPv6, and objects for use on IPv6 routers. In short, it is not expected that every entity will implement all of the objects within this MIB. The reader should consult the conformance and compliance section to determine which objects are appropriate for a given entity.

3.2.1. General Objects

In both IPv4 and IPv6, there are only a small number of "knobs" for controlling the general IP stack. Most controls will be in a more specific setting, such as for controlling a router or TCP engine.

This MIB defines a total of three general knobs, only two of which are used for both IPv4 and IPv6.

Objects are included for both protocols to enable or disable forwarding and to set limits on the lifetime of a packet (ttl or hop count).

The third knob, the timeout period for reassembling fragments, is only defined for IPv4, as IPv6 specifies this value directly.

Each group of objects is required when implementing their respective protocols.

3.2.2. Interface Tables

This MIB includes a pair of tables to convey information about the IPv4 and IPv6 protocols that is interface specific.

Special note should be taken of the administrative status objects. These are defined to allow each protocol to selectively enable or disable interfaces. These objects can be used in conjunction with the ifAdminStatus object to manipulate the interfaces as necessary. With these three objects, an interface may be enabled or disabled completely, as well as connected to the IPv4 stack, the IPv6 stack or both stacks. Setting ifAdminStatus to "down" should not affect the protocol specific status objects.

Each interface table is required when implementing their respective protocols.

3.2.3. IP Statistics Tables

The IP statistics tables (ipSystemStatsTable and ipIfStatsTable) contain objects to count the number of datagrams and octets that a given entity has processed. Unlike the previous attempt, this document uses a single table for multiple address types. Typically the only two types of interest are IPv4 and IPv6; however, the table can support other types if necessary.

The first table, ipSystemStatsTable, conveys system wide information. (That is, the various counters are for all interfaces and not a specific set of interfaces.) Its index is formed from a single

sub-id that represents the address type for which the statistics were counted.

The second table, `ipIfStatsTable`, conveys interface specific information. Its index is formed from two sub-ids. The first represents the address type (IPv4 and IPv6), and the interface within that address type is represented by the second sub-id.

The two tables have a similar set of objects that are intended to count the same things, except for the difference in granularity. The object ID `"ipSystemStatsEntry.2"` is reserved in order to align the object IDs of the counters in the first table with their counterparts in the second table.

Several objects to note are `ipSystemStatsDiscontinuityTime`, `ipIfStatsDiscontinuityTime`, `ipSystemStatsRefreshRate`, and `ipIfStatsRefreshRate`. These objects provide information about the row in the table more than about the system itself.

The discontinuity objects allow a management entity to determine if a discontinuity event that would invalidate the management entity's understanding of the counters has occurred. The system being re-initialized or the interface being cycled are possible examples of a discontinuity event.

The refresh objects allow a management entity to determine a proper polling interval for the rest of the objects.

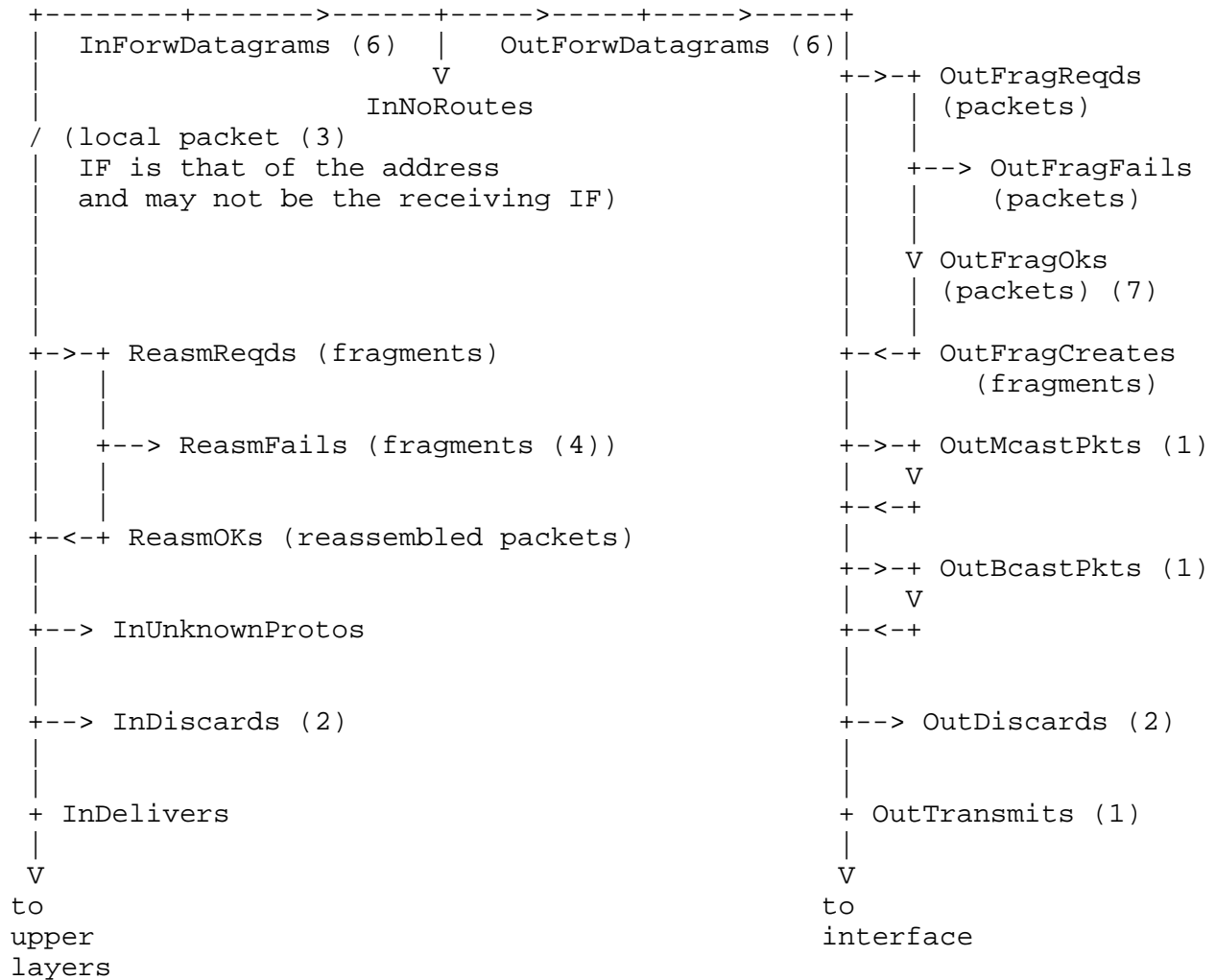
The following Case diagram represents the general ordering of the packet counters. In order to avoid extra clutter, the prefixes `"ipSystemStats"` and `"ipIfStats"` have been removed from each of the counter names.

from
interface

```
V
|
+ InReceives (1)
|
+--> InHdrErrors (5)
|
+-->+ InMcastPkts (1)
|   V
+--<+
|
+-->+ InBcastPkts (1)
|   V
+--<+
|
+--> InTruncatedPkts (5)
|
+--> InAddrErrors
|
+--> InDiscards (2)
|
```

from
upper
layers

```
V
|
+ OutRequests
|
+--> OutNoRoutes
|
```



- (1) The HC counters and octet counters are also found at these points but have been left out for clarity.
- (2) The discard counters may increment at any time in the processing path. Packets discarded to the left of InNoRoutes cause the InDiscards counter to increment, while those discarded to the right are counted in the OutDiscards counters.
- (3) Local packets on the input side are counted on the interface associated with their destination address, which may not be the interface on which they were received. This requirement is caused by the possibility of losing the original interface during processing, especially re-assembly.

- (4) Some re-assembly algorithms may lose track of the number of fragments during processing and so some fragments may not be counted in this object.
- (5) InTruncatedPkts should only be incremented if the frame contained a valid header but was otherwise shorter than required. Frames that are too short to contain a valid header should be counted as InHdrErrors.
- (6) The forwarding objects may be incremented, even for packets that originated locally or are destined for the local host, if their addresses are such that the local host would need to forward the packet to pass it to the correct interface.
- (7) When fragmenting a packet, an entity should increment the OutFragFails counter, rather than the OutDiscards counter, in order to preserve the equation $\text{FragOks} + \text{FragFails} == \text{FragRqds}$.

The objects in both tables are spread amongst several conformance groups based on the bandwidth required to wrap the counters within an hour. The base system group is mandatory for all entities. The other system groups are optional depending on bandwidth. The interface specific-groups are optional.

3.2.4. Internet Address Prefix Table

This table provides information about the prefixes this entity is using, including their lifetimes. This table provides a convenient place to which other tables that make use of prefixes, such as the ipAddressTable, may point. By including this table, the MIB can supply the prefix information for all addresses, yet minimize the amount of duplication required in storing and accessing this data. This arrangement also clarifies the relationship between addresses that have the same prefix.

This table is required for IPv6 entities.

3.2.5. Internet Address Table

This table lists the IP addresses (both IPv4 and IPv6) used by this entity. It also includes some basic information about how and when the address was formed and last updated. This table allows a manager to determine who a given entity thinks it is.

This table is required for all IP entities.

3.2.6. Internet Address Translation Table

This table provides a mapping between IP layer addresses and physical addresses as would be formed by either Address Resolution Protocol (ARP) for IPv4 or the neighbor discovery protocol for IPv6.

3.2.7. IPv6 Scope Zone Index Table

This table specifies the zone index to interface mapping. By examining the table, a manager can determine which groups of interfaces are within a particular zone for a given scope.

The zone index information is only valid within a given entity; the indexes used on one entity may not be comparable to those used on a different entity.

This table is required for IPv6 entities.

3.2.8. Default Router Table

This table lists the default routers known to this entity. This table is intended to be a simple list to display the information that end nodes may have been configured with or acquired through a simple system such as IPv6 router advertisements. Managers attempting to view more complicated routing information should examine the routing specific tables from other MIBs.

This table is required for all entities.

3.2.9. Router Advertisement Table

This table contains the non-routing information that an IPv6 router would use in constructing a router advertisement message. It does not contain information about the prefixes or other routing specific information that the router might advertise. The router should acquire such information from either the routing tables or from some routing table specific MIB.

This table is only required for IPv6 router entities.

3.2.10. ICMP Statistics Tables

There are two sets of statistics for ICMP. The first contains a simple set of counters to track the number of ICMP messages and errors processed by this entity.

The second supplies more detail about the ICMP messages processed by this entity. Its index is formed from two sub-ids. The first represents the address type (IPv4 and IPv6), and the second represents the particular message type being counted. A given row need not be instantiated unless a message of that type has been processed, i.e., the row for icmpMsgStatsType=X MAY be instantiated before but MUST be instantiated after the first message with Type=X is received or transmitted. After receiving or transmitting any succeeding messages with Type=X, the relevant counter must be incremented.

Both of these tables are required for all entities.

3.2.11. Conformance and Compliance

This MIB contains several sets of objects. Some of these sets are useful on all types of entities, while others are only useful on a limited subset of entities. The conformance section attempts to group the objects into sets that may be discussed as units, and the compliance section then details which of these units are required in various circumstances.

The circumstances used in the compliance section are implementing IPv4, IPv6, or IPv6 router functions and having a bandwidth of less than 20MB, between 20MB and 650MB, or greater than 650MB.

3.2.12. Deprecated Objects

This MIB also includes a set of deprecated objects from previous iterations. They are included as part of the historical record.

4. Updating Implementations

There are several general classes of change that are required.

The first and most major change is that most of the previous objects have different object IDs and additional indexes to support the possibility of different address types. The general counters for IP and ICMP are examples of this. They have been moved to the ipSystemStatsTable and icmpMsgStatsTable, respectively.

The second change is the extension of all address objects to allow for both IPv4 and IPv6 addresses and the addition of an address type object to specify what address type is in use.

The third change is the addition of several new objects to the replacement for a previously existing table such as ipNetToPhysical.

The fourth change is the addition of completely new tables such as `ipIfStatsTable` and `ipDefaultRouterTable`. The first is based on the previous statistics groups, while the second is completely new to this MIB.

4.1. Updating an Implementation of the IPv4-only IP-MIB

The somewhat more specific changes that are required for IPv4 follow. Note well: this is not meant to be an exhaustive list and the reader should examine the MIB for full details.

Several of the general objects (`ipForwarding`, `ipDefaultTTL`, `ipReasmTimeout`) remain unchanged.

Most of the rest of the general objects were counters and have been moved into the `ipSystemStatsTable`. The basic instrumentation should remain the same, though the object definitions should be checked for clarifications. If they aren't already in a structure, putting the counter variables in one would be useful. Several new objects have been added to count additional items, and instrumentation code must be added for these objects. Finally, the SNMP routines must be updated to handle the new indexing.

In addition to the `ipSystemStatsTable`, the MIB includes the `ipIfStatsTable`. This table counts the same items as the system table but does so on a per interface basis. It is optional and may be ignored. If you decide to implement it, you may wish to arrange to collect the data on a per-interface basis and then sum those counters in order to provide the aggregate system level statistics. However, if you choose to provide the system level statistics by summing the interface level counters, no interface level statistics can be lost - if an interface is removed, the statistics associated with it must be retained.

The `ipAddrTable` has, loosely, been converted to the `ipAddressTable`. While the general idea remains the same, the `ipAddressTable` is sufficiently different that writing new code may be easier than updating old code. The primary difference is the addition of several new objects. In addition, the `ipAdEntReasmMaxSize` has been moved to another table, `ipv4InterfaceTable`. As above, the SNMP routines will need to be updated to handle the new indexing.

The `ipNetToMediaTable` has been moved to the `ipNetToPhysicalTable`. These tables are fairly similar and updating the old code may be straightforward. As above, the SNMP routines will need to be updated to handle the new indexing.

Two new tables, `ipv4InterfaceTable` and `ipDefaultRouterTable`, are required as well as several new ICMP counters.

Finally, there are several tables that are required for IPv6 but are optional for IPv4 that you may elect to implement.

4.2. Updating an Implementation of the IPv6-MIB

The somewhat more specific changes that are required for IPv6 follow. Note well: this is not meant to be an exhaustive list and the reader should examine the MIB for full details.

Two of the general objects, `ipv6Forwarding` and `ipv6DefaultHopLimit`, have been renamed and given new object identifiers within the `ip` branch but are otherwise unchanged. The new names are `ipv6IpForwarding` and `ipv6IpDefaultHopLimit`.

While there is an `ipv6InterfaceTable` that contains some of the pieces from the `ipv6IfTable`, the two are somewhat different in concept. The `ipv6IfTable` was meant to replicate the `ifTable` while the `ipv6InterfaceTable` is meant to be an addition to the `ifTable`. As such, items that were duplicated between the `ifTable` and `ipv6IfTable` have been removed and some new objects added.

The `ipv6IfStatsTable` most closely resembles the `ipIfStatsTable` with an additional index for the address type and most of the instrumentation should be re-usable. Some new objects have been added to the `ipIfStatsTable`. As above, the SNMP routines will need to be updated to handle the new indexing. Finally, the `ipIfStatsTable` is optional and may be ignored.

The `ipSystemStatsTable` is effectively new, but it may be able to make use of most of the instrumentation from the old `ipv6IfStatsTable`. As with the IPv4 discussion, one implementation strategy would be to count the statistics for the `ipIfStatsTable` and aggregate them when queried for this table. Again, as with the IPv4 discussion, this strategy only works if the interfaces cannot be removed or if the statistics for removed interfaces are somehow retained.

The `ipv6AddrPrefixTable` is now the `ipAddressPrefixTable`. The new table contains an extra object and the additional index required for IPv4 compatibility. As above, the SNMP routines will need to be updated to handle the new indexing.

The `ipAddressTable` is loosely based on the `ipv6AddrTable` but has changed considerably with the addition of several new objects and the removal of one of its indexes.

The IPv6 routing information (ipv6RouteNumber, ipv6DiscardedRoutes, and ipv6RouteTable) has been removed from this MIB. The replacements or updates for this information is in the update to the IP Forwarding Table MIB [16]. The ipv6NetToMediaTable has been converted to the ipNetToPhysicalTable. The new table contains an extra object and the additional index required for IPv4 compatibility. As above, the SNMP routines will need to be updated to handle the new indexing.

The ICMP tables have been substantially changed. The previous tables required counting on a per-message and per-interface basis. The new tables only require counting on a per-message, per-protocol basis and include an aggregate of all messages on a per-protocol basis.

In addition to the above, several new tables have been added. Both the ipv6ScopeZoneIndexTable and ipDefaultRouterTable are required on all IPv6 entities. The ipv6RouterAdvertTable is only required on IPv6 routers.

5. Definitions

The following MIB module imports from the IF-MIB [6] and the INET-ADDRESS-MIB [7] and references Neighbor Discovery [4], the IPv6 Stateless Address Autoconfiguration protocol [5], the Default Router Preferences document [8], ARP [10] and the IPv6 address architecture document [17].

IP-MIB DEFINITIONS ::= BEGIN

IMPORTS

```

MODULE-IDENTITY, OBJECT-TYPE,
Integer32, Counter32, IpAddress,
mib-2, Unsigned32, Counter64,
zeroDotZero                                FROM SNMPv2-SMI
PhysAddress, TruthValue,
TimeStamp, RowPointer,
TEXTUAL-CONVENTION, TestAndIncr,
RowStatus, StorageType                     FROM SNMPv2-TC
MODULE-COMPLIANCE, OBJECT-GROUP            FROM SNMPv2-CONF
InetAddress, InetAddressType,
InetAddressPrefixLength,
InetVersion, InetZoneIndex                 FROM INET-ADDRESS-MIB
InterfaceIndex                             FROM IF-MIB;
```

ipMIB MODULE-IDENTITY

```

LAST-UPDATED "200602020000Z"
ORGANIZATION "IETF IPv6 MIB Revision Team"
CONTACT-INFO
    "Editor:
```

Shawn A. Routhier
Interworking Labs
108 Whispering Pines Dr. Suite 235
Scotts Valley, CA 95066
USA
EMail: <sar@iwl.com>

DESCRIPTION

"The MIB module for managing IP and ICMP implementations, but excluding their management of IP routes.

Copyright (C) The Internet Society (2006). This version of this MIB module is part of RFC 4293; see the RFC itself for full legal notices."

REVISION "200602020000Z"

DESCRIPTION

"The IP version neutral revision with added IPv6 objects for ND, default routers, and router advertisements. As well as being the successor to RFC 2011, this MIB is also the successor to RFCs 2465 and 2466. Published as RFC 4293."

REVISION "199411010000Z"

DESCRIPTION

"A separate MIB module (IP-MIB) for IP and ICMP management objects. Published as RFC 2011."

REVISION "199103310000Z"

DESCRIPTION

"The initial revision of this MIB module was part of MIB-II, which was published as RFC 1213."

::= { mib-2 48}

--

-- The textual conventions we define and use in this MIB.

--

IpAddressOriginTC ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"The origin of the address.

manual(2) indicates that the address was manually configured to a specified address, e.g., by user configuration.

dhcp(4) indicates an address that was assigned to this system by a DHCP server.

linklayer(5) indicates an address created by IPv6 stateless

auto-configuration.

random(6) indicates an address chosen by the system at random, e.g., an IPv4 address within 169.254/16, or an RFC 3041 privacy address."

```
SYNTAX      INTEGER {  
    other(1),  
    manual(2),  
    dhcp(4),  
    linklayer(5),  
    random(6)  
}
```

IpAddressStatusTC ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"The status of an address. Most of the states correspond to states from the IPv6 Stateless Address Autoconfiguration protocol.

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 isn't a valid address and it shouldn't 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.

The unknown(5) state indicates that the status cannot be determined for some reason.

The tentative(6) state indicates that the uniqueness of the address on the link is being verified. Addresses in this state should not be used for general communication and should only be used to determine the uniqueness of the address.

The duplicate(7) state indicates the address has been determined to be non-unique on the link and so must not be

used.

The optimistic(8) state indicates the address is available for use, subject to restrictions, while its uniqueness on a link is being verified.

In the absence of other information, an IPv4 address is always preferred(1)."

REFERENCE "RFC 2462"

```
SYNTAX      INTEGER {
    preferred(1),
    deprecated(2),
    invalid(3),
    inaccessible(4),
    unknown(5),
    tentative(6),
    duplicate(7),
    optimistic(8)
}
```

IpAddressPrefixOriginTC ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"The origin of this prefix.

manual(2) indicates a prefix that was manually configured.

wellknown(3) indicates a well-known prefix, e.g., 169.254/16 for IPv4 auto-configuration or fe80::/10 for IPv6 link-local addresses. Well known prefixes may be assigned by IANA, the address registries, or by specification in a standards track RFC.

dhcp(4) indicates a prefix that was assigned by a DHCP server.

routeradv(5) indicates a prefix learned from a router advertisement.

Note: while IpAddressOriginTC and IpAddressPrefixOriginTC are similar, they are not identical. The first defines how an address was created, while the second defines how a prefix was found."

```
SYNTAX      INTEGER {
    other(1),
    manual(2),
    wellknown(3),
    dhcp(4),
}
```

```

        routeradv(5)
    }

Ipv6AddressIfIdentifierTC ::= 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))

--
-- the IP general group
-- some objects that affect all of IPv4
--

ip          OBJECT IDENTIFIER ::= { mib-2 4 }

ipForwarding 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 IPv4
        router in respect to the forwarding of datagrams received
        by, but not addressed to, this entity. IPv4 routers forward
        datagrams. IPv4 hosts do not (except those source-routed
        via the host).

        When this object is written, the entity should save the
        change to non-volatile storage and restore the object from
        non-volatile storage upon re-initialization of the system.
        Note: a stronger requirement is not used because this object
        was previously defined."
    ::= { ip 1 }

ipDefaultTTL OBJECT-TYPE
    SYNTAX      Integer32 (1..255)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The default value inserted into the Time-To-Live field of
        the IPv4 header of datagrams originated at this entity,
        whenever a TTL value is not supplied by the transport layer

```

protocol.

When this object is written, the entity should save the change to non-volatile storage and restore the object from non-volatile storage upon re-initialization of the system.

Note: a stronger requirement is not used because this object was previously defined."

```
::= { ip 2 }
```

ipReasmTimeout OBJECT-TYPE

SYNTAX Integer32

UNITS "seconds"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The maximum number of seconds that received fragments are held while they are awaiting reassembly at this entity."

```
::= { ip 13 }
```

--

-- the IPv6 general group

-- Some objects that affect all of IPv6

--

ipv6IpForwarding 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 on any interface 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).

When this object is written, the entity SHOULD save the change to non-volatile storage and restore the object from non-volatile storage upon re-initialization of the system."

```
::= { ip 25 }
```

ipv6IpDefaultHopLimit OBJECT-TYPE

SYNTAX Integer32 (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.

When this object is written, the entity SHOULD save the change to non-volatile storage and restore the object from non-volatile storage upon re-initialization of the system."

REFERENCE "RFC 2461 Section 6.3.2"

::= { ip 26 }

--

-- IPv4 Interface Table

--

ipv4InterfaceTableLastChange OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of sysUpTime on the most recent occasion at which a row in the ipv4InterfaceTable was added or deleted, or when an ipv4InterfaceReasmMaxSize or an ipv4InterfaceEnableStatus object was modified.

If new objects are added to the ipv4InterfaceTable that require the ipv4InterfaceTableLastChange to be updated when they are modified, they must specify that requirement in their description clause."

::= { ip 27 }

ipv4InterfaceTable OBJECT-TYPE

SYNTAX SEQUENCE OF Ipv4InterfaceEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The table containing per-interface IPv4-specific information."

::= { ip 28 }

ipv4InterfaceEntry OBJECT-TYPE

SYNTAX Ipv4InterfaceEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry containing IPv4-specific information for a specific interface."

INDEX { ipv4InterfaceIfIndex }

```
::= { ipv4InterfaceTable 1 }
```

```
Ipv4InterfaceEntry ::= SEQUENCE {
    ipv4InterfaceIfIndex      InterfaceIndex,
    ipv4InterfaceReasmMaxSize Integer32,
    ipv4InterfaceEnableStatus INTEGER,
    ipv4InterfaceRetransmitTime Unsigned32
}
```

```
ipv4InterfaceIfIndex OBJECT-TYPE
```

```
SYNTAX      InterfaceIndex
```

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

```
STATUS      current
```

```
DESCRIPTION
```

"The index value that uniquely identifies the interface to which this entry is applicable. The interface identified by a particular value of this index is the same interface as identified by the same value of the IF-MIB's ifIndex."

```
::= { ipv4InterfaceEntry 1 }
```

```
ipv4InterfaceReasmMaxSize OBJECT-TYPE
```

```
SYNTAX      Integer32 (0..65535)
```

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

```
STATUS      current
```

```
DESCRIPTION
```

"The size of the largest IPv4 datagram that this entity can re-assemble from incoming IPv4 fragmented datagrams received on this interface."

```
::= { ipv4InterfaceEntry 2 }
```

```
ipv4InterfaceEnableStatus OBJECT-TYPE
```

```
SYNTAX      INTEGER {
                up(1),
                down(2)
            }
```

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

```
STATUS      current
```

```
DESCRIPTION
```

"The indication of whether IPv4 is enabled (up) or disabled (down) on this interface. This object does not affect the state of the interface itself, only its connection to an IPv4 stack. The IF-MIB should be used to control the state of the interface."

```
::= { ipv4InterfaceEntry 3 }
```

```
ipv4InterfaceRetransmitTime OBJECT-TYPE
```

```
SYNTAX      Unsigned32
```

```
UNITS      "milliseconds"
```

MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "The time between retransmissions of ARP requests to a
 neighbor when resolving the address or when probing the
 reachability of a neighbor."
REFERENCE "RFC 1122"
DEFVAL { 1000 }
::= { ipv4InterfaceEntry 4 }

--
-- v6 interface table
--

ipv6InterfaceTableLastChange OBJECT-TYPE

SYNTAX TimeStamp
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "The value of sysUpTime on the most recent occasion at which
 a row in the ipv6InterfaceTable was added or deleted or when
 an ipv6InterfaceReasmMaxSize, ipv6InterfaceIdentifier,
 ipv6InterfaceEnableStatus, ipv6InterfaceReachableTime,
 ipv6InterfaceRetransmitTime, or ipv6InterfaceForwarding
 object was modified.

 If new objects are added to the ipv6InterfaceTable that
 require the ipv6InterfaceTableLastChange to be updated when
 they are modified, they must specify that requirement in
 their description clause."
::= { ip 29 }

ipv6InterfaceTable OBJECT-TYPE

SYNTAX SEQUENCE OF Ipv6InterfaceEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
 "The table containing per-interface IPv6-specific
 information."
::= { ip 30 }

ipv6InterfaceEntry OBJECT-TYPE

SYNTAX Ipv6InterfaceEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
 "An entry containing IPv6-specific information for a given
 interface."

```

INDEX { ipv6InterfaceIfIndex }
 ::= { ipv6InterfaceTable 1 }

```

```

Ipv6InterfaceEntry ::= SEQUENCE {
    ipv6InterfaceIfIndex          InterfaceIndex,
    ipv6InterfaceReasmMaxSize     Unsigned32,
    ipv6InterfaceIdentifier       Ipv6AddressIfIdentifierTC,
    ipv6InterfaceEnableStatus     INTEGER,
    ipv6InterfaceReachableTime    Unsigned32,
    ipv6InterfaceRetransmitTime   Unsigned32,
    ipv6InterfaceForwarding       INTEGER
}

```

ipv6InterfaceIfIndex OBJECT-TYPE

SYNTAX InterfaceIndex

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The index value that uniquely identifies the interface to which this entry is applicable. The interface identified by a particular value of this index is the same interface as identified by the same value of the IF-MIB's ifIndex."

```
 ::= { ipv6InterfaceEntry 1 }
```

ipv6InterfaceReasmMaxSize OBJECT-TYPE

SYNTAX Unsigned32 (1500..65535)

UNITS "octets"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

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

```
 ::= { ipv6InterfaceEntry 2 }
```

ipv6InterfaceIdentifier OBJECT-TYPE

SYNTAX Ipv6AddressIfIdentifierTC

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The Interface Identifier for this interface. The Interface Identifier is combined with an address prefix to form an interface address."

By default, the Interface Identifier is auto-configured according to the rules of the link type to which this interface is attached.

A zero length identifier may be used where appropriate. One possible example is a loopback interface."

```
::= { ipv6InterfaceEntry 3 }
```

```
-- This object ID is reserved as it was used in earlier versions of
-- the MIB module. In theory, OIDs are not assigned until the
-- specification is released as an RFC; however, as some companies
-- may have shipped code based on earlier versions of the MIB, it
-- seems best to reserve this OID. This OID had been
-- ipv6InterfacePhysicalAddress.
-- ::= { ipv6InterfaceEntry 4}
```

ipv6InterfaceEnableStatus OBJECT-TYPE

```
SYNTAX      INTEGER {
                up(1),
                down(2)
            }
```

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

```
STATUS      current
```

```
DESCRIPTION
```

"The indication of whether IPv6 is enabled (up) or disabled (down) on this interface. This object does not affect the state of the interface itself, only its connection to an IPv6 stack. The IF-MIB should be used to control the state of the interface.

When this object is written, the entity SHOULD save the change to non-volatile storage and restore the object from non-volatile storage upon re-initialization of the system."

```
::= { ipv6InterfaceEntry 5 }
```

ipv6InterfaceReachableTime OBJECT-TYPE

```
SYNTAX      Unsigned32
```

```
UNITS       "milliseconds"
```

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

```
STATUS      current
```

```
DESCRIPTION
```

"The time a neighbor is considered reachable after receiving a reachability confirmation."

```
REFERENCE "RFC 2461, Section 6.3.2"
```

```
::= { ipv6InterfaceEntry 6 }
```

ipv6InterfaceRetransmitTime OBJECT-TYPE

```
SYNTAX      Unsigned32
```

```
UNITS       "milliseconds"
```

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

```
STATUS      current
```

```
DESCRIPTION
```

"The time between retransmissions of Neighbor Solicitation messages to a neighbor when resolving the address or when probing the reachability of a neighbor."

REFERENCE "RFC 2461, Section 6.3.2"

::= { ipv6InterfaceEntry 7 }

ipv6InterfaceForwarding 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 on this interface with 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).

This object is constrained by ipv6IpForwarding and is ignored if ipv6IpForwarding is set to notForwarding. Those systems that do not provide per-interface control of the forwarding function should set this object to forwarding for all interfaces and allow the ipv6IpForwarding object to control the forwarding capability.

When this object is written, the entity SHOULD save the change to non-volatile storage and restore the object from non-volatile storage upon re-initialization of the system."

::= { ipv6InterfaceEntry 8 }

--

-- Per-Interface or System-Wide IP statistics.

--

-- The following two tables, ipSystemStatsTable and ipIfStatsTable, are intended to provide the same counters at different granularities. The ipSystemStatsTable provides system wide counters aggregating the traffic counters for all interfaces for a given address type. The ipIfStatsTable provides the same counters but for specific interfaces rather than as an aggregate.

--

-- Note well: If a system provides both system-wide and interface-specific values, the system-wide value may not be equal to the sum of the interface-specific values across all interfaces due to e.g., dynamic interface creation/deletion.

--

-- Note well: Both of these tables contain some items that are

```
-- represented by two objects, representing the value in either 32
-- or 64 bits.  For those objects, the 32-bit value MUST be the low
-- order 32 bits of the 64-bit value.  Also note that the 32-bit
-- counters must be included when the 64-bit counters are included.
```

```
ipTrafficStats OBJECT IDENTIFIER ::= { ip 31 }
```

```
ipSystemStatsTable OBJECT-TYPE
```

```
    SYNTAX      SEQUENCE OF IpSystemStatsEntry
```

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

```
    STATUS      current
```

```
    DESCRIPTION
```

```
        "The table containing system wide, IP version specific
        traffic statistics.  This table and the ipIfStatsTable
        contain similar objects whose difference is in their
        granularity.  Where this table contains system wide traffic
        statistics, the ipIfStatsTable contains the same statistics
        but counted on a per-interface basis."
```

```
    ::= { ipTrafficStats 1 }
```

```
ipSystemStatsEntry OBJECT-TYPE
```

```
    SYNTAX      IpSystemStatsEntry
```

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

```
    STATUS      current
```

```
    DESCRIPTION
```

```
        "A statistics entry containing system-wide objects for a
        particular IP version."
```

```
    INDEX { ipSystemStatsIPVersion }
```

```
    ::= { ipSystemStatsTable 1 }
```

```
IpSystemStatsEntry ::= SEQUENCE {
```

ipSystemStatsIPVersion	InetVersion,
ipSystemStatsInReceives	Counter32,
ipSystemStatsHCInReceives	Counter64,
ipSystemStatsInOctets	Counter32,
ipSystemStatsHCInOctets	Counter64,
ipSystemStatsInHdrErrors	Counter32,
ipSystemStatsInNoRoutes	Counter32,
ipSystemStatsInAddrErrors	Counter32,
ipSystemStatsInUnknownProtos	Counter32,
ipSystemStatsInTruncatedPkts	Counter32,
ipSystemStatsInForwDatagrams	Counter32,
ipSystemStatsHCInForwDatagrams	Counter64,
ipSystemStatsReasmReqds	Counter32,
ipSystemStatsReasmOKs	Counter32,
ipSystemStatsReasmFails	Counter32,
ipSystemStatsInDiscards	Counter32,
ipSystemStatsInDelivers	Counter32,

```

    ipSystemStatsHCInDelivers      Counter64,
    ipSystemStatsOutRequests      Counter32,
    ipSystemStatsHCOutRequests    Counter64,
    ipSystemStatsOutNoRoutes      Counter32,
    ipSystemStatsOutForwDatagrams Counter32,
    ipSystemStatsHCOutForwDatagrams Counter64,
    ipSystemStatsOutDiscards      Counter32,
    ipSystemStatsOutFragReqds     Counter32,
    ipSystemStatsOutFragOKs       Counter32,
    ipSystemStatsOutFragFails     Counter32,
    ipSystemStatsOutFragCreates   Counter32,
    ipSystemStatsOutTransmits     Counter32,
    ipSystemStatsHCOutTransmits   Counter64,
    ipSystemStatsOutOctets        Counter32,
    ipSystemStatsHCOutOctets      Counter64,
    ipSystemStatsInMcastPkts      Counter32,
    ipSystemStatsHCInMcastPkts    Counter64,
    ipSystemStatsInMcastOctets    Counter32,
    ipSystemStatsHCInMcastOctets  Counter64,
    ipSystemStatsOutMcastPkts     Counter32,
    ipSystemStatsHCOutMcastPkts   Counter64,
    ipSystemStatsOutMcastOctets   Counter32,
    ipSystemStatsHCOutMcastOctets Counter64,
    ipSystemStatsInBcastPkts      Counter32,
    ipSystemStatsHCInBcastPkts    Counter64,
    ipSystemStatsOutBcastPkts     Counter32,
    ipSystemStatsHCOutBcastPkts   Counter64,
    ipSystemStatsDiscontinuityTime Timestamp,
    ipSystemStatsRefreshRate      Unsigned32
}

```

ipSystemStatsIPVersion OBJECT-TYPE

```

SYNTAX      InetVersion
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "The IP version of this row."
 ::= { ipSystemStatsEntry 1 }

```

```

-- This object ID is reserved to allow the IDs for this table's objects
-- to align with the objects in the ipIfStatsTable.
-- ::= { ipSystemStatsEntry 2 }

```

ipSystemStatsInReceives OBJECT-TYPE

```

SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION

```

"The total number of input IP datagrams received, including those received in error.

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

::= { ipSystemStatsEntry 3 }

ipSystemStatsHCInReceives OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of input IP datagrams received, including those received in error. This object counts the same datagrams as ipSystemStatsInReceives, but allows for larger values.

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

::= { ipSystemStatsEntry 4 }

ipSystemStatsInOctets OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of octets received in input IP datagrams, including those received in error. Octets from datagrams counted in ipSystemStatsInReceives MUST be counted here.

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

::= { ipSystemStatsEntry 5 }

ipSystemStatsHCInOctets OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of octets received in input IP datagrams, including those received in error. This object counts the same octets as ipSystemStatsInOctets, but allows for larger

values.

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

::= { ipSystemStatsEntry 6 }

ipSystemStatsInHdrErrors OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of input IP datagrams discarded due to errors in their IP headers, including version number mismatch, other format errors, hop count exceeded, errors discovered in processing their IP options, etc.

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

::= { ipSystemStatsEntry 7 }

ipSystemStatsInNoRoutes OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

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

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

::= { ipSystemStatsEntry 8 }

ipSystemStatsInAddrErrors OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

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

datagrams, this counter includes datagrams discarded because the destination address was not a local address.

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

::= { ipSystemStatsEntry 9 }

ipSystemStatsInUnknownProtos OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of locally-addressed IP datagrams received successfully but discarded because of an unknown or unsupported protocol.

When tracking interface statistics, the counter of the interface to which these datagrams were addressed is incremented. This interface might not be the same as the input interface for some of the datagrams.

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

::= { ipSystemStatsEntry 10 }

ipSystemStatsInTruncatedPkts OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

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

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

::= { ipSystemStatsEntry 11 }

ipSystemStatsInForwDatagrams OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of input datagrams for which this entity was not their final IP destination and for which this entity attempted to find a route to forward them to that final destination. In entities that do not act as IP routers, this counter will include only those datagrams that were Source-Routed via this entity, and the Source-Route processing was successful.

When tracking interface statistics, the counter of the incoming interface is incremented for each datagram.

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

::= { ipSystemStatsEntry 12 }

ipSystemStatsHCInForwDatagrams OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of input datagrams for which this entity was not their final IP destination and for which this entity attempted to find a route to forward them to that final destination. This object counts the same packets as ipSystemStatsInForwDatagrams, but allows for larger values.

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

::= { ipSystemStatsEntry 13 }

ipSystemStatsReasmReqds OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of IP fragments received that needed to be reassembled at this interface.

When tracking interface statistics, the counter of the interface to which these fragments were addressed is incremented. This interface might not be the same as the input interface for some of the fragments.

Discontinuities in the value of this counter can occur at

re-initialization of the management system, and at other times as indicated by the value of ipSystemStatsDiscontinuityTime."

```
::= { ipSystemStatsEntry 14 }
```

ipSystemStatsReasmOKs OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"The number of IP datagrams successfully reassembled.

When tracking interface statistics, the counter of the interface to which these datagrams were addressed is incremented. This interface might not be the same as the input interface for some of the datagrams.

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

```
::= { ipSystemStatsEntry 15 }
```

ipSystemStatsReasmFails OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"The number of failures detected by the IP re-assembly algorithm (for whatever reason: timed out, errors, etc.). Note that this is not necessarily a count of discarded IP 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.

When tracking interface statistics, the counter of the interface to which these fragments were addressed is incremented. This interface might not be the same as the input interface for some of the fragments.

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

```
::= { ipSystemStatsEntry 16 }
```

ipSystemStatsInDiscards OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only
STATUS current
DESCRIPTION

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

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

::= { ipSystemStatsEntry 17 }

ipSystemStatsInDelivers OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"The total number of datagrams successfully delivered to IP user-protocols (including ICMP).

When tracking interface statistics, the counter of the interface to which these datagrams were addressed is incremented. This interface might not be the same as the input interface for some of the datagrams.

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

::= { ipSystemStatsEntry 18 }

ipSystemStatsHCInDelivers OBJECT-TYPE

SYNTAX Counter64
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"The total number of datagrams successfully delivered to IP user-protocols (including ICMP). This object counts the same packets as ipSystemStatsInDelivers, but allows for larger values.

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

```
::= { ipSystemStatsEntry 19 }
```

ipSystemStatsOutRequests OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of IP datagrams that local IP user-protocols (including ICMP) supplied to IP in requests for transmission. Note that this counter does not include any datagrams counted in ipSystemStatsOutForwDatagrams.

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

```
::= { ipSystemStatsEntry 20 }
```

ipSystemStatsHCOutRequests OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of IP datagrams that local IP user-protocols (including ICMP) supplied to IP in requests for transmission. This object counts the same packets as ipSystemStatsOutRequests, but allows for larger values.

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

```
::= { ipSystemStatsEntry 21 }
```

ipSystemStatsOutNoRoutes OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of locally generated IP datagrams discarded because no route could be found to transmit them to their destination.

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

```
::= { ipSystemStatsEntry 22 }
```

`ipSystemStatsOutForwDatagrams OBJECT-TYPE``SYNTAX Counter32``MAX-ACCESS read-only``STATUS current``DESCRIPTION`

"The number of datagrams for which this entity was not their final IP destination and for which it was successful in finding a path to their final destination. In entities that do not act as IP routers, this counter will include only those datagrams that were Source-Routed via this entity, and the Source-Route processing was successful.

When tracking interface statistics, the counter of the outgoing interface is incremented for a successfully forwarded datagram.

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

`::= { ipSystemStatsEntry 23 }``ipSystemStatsHCOutForwDatagrams OBJECT-TYPE``SYNTAX Counter64``MAX-ACCESS read-only``STATUS current``DESCRIPTION`

"The number of datagrams for which this entity was not their final IP destination and for which it was successful in finding a path to their final destination. This object counts the same packets as `ipSystemStatsOutForwDatagrams`, but allows for larger values.

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

`::= { ipSystemStatsEntry 24 }``ipSystemStatsOutDiscards OBJECT-TYPE``SYNTAX Counter32``MAX-ACCESS read-only``STATUS current``DESCRIPTION`

"The number of output IP datagrams for which no problem was encountered to prevent their transmission to their destination, but were discarded (e.g., for lack of buffer space). Note that this counter would include

datagrams counted in ipSystemStatsOutForwDatagrams if any such datagrams met this (discretionary) discard criterion.

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

::= { ipSystemStatsEntry 25 }

ipSystemStatsOutFragReqds OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of IP datagrams that would require fragmentation in order to be transmitted.

When tracking interface statistics, the counter of the outgoing interface is incremented for a successfully fragmented datagram.

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

::= { ipSystemStatsEntry 26 }

ipSystemStatsOutFragOKs OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of IP datagrams that have been successfully fragmented.

When tracking interface statistics, the counter of the outgoing interface is incremented for a successfully fragmented datagram.

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

::= { ipSystemStatsEntry 27 }

ipSystemStatsOutFragFails OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of IP datagrams that have been discarded because they needed to be fragmented but could not be. This includes IPv4 packets that have the DF bit set and IPv6 packets that are being forwarded and exceed the outgoing link MTU.

When tracking interface statistics, the counter of the outgoing interface is incremented for an unsuccessfully fragmented datagram.

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

::= { ipSystemStatsEntry 28 }

ipSystemStatsOutFragCreates 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 IP fragmentation.

When tracking interface statistics, the counter of the outgoing interface is incremented for a successfully fragmented datagram.

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

::= { ipSystemStatsEntry 29 }

ipSystemStatsOutTransmits OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of IP datagrams that this entity supplied to the lower layers for transmission. This includes datagrams generated locally and those forwarded by 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 the value of
ipSystemStatsDiscontinuityTime."
 ::= { ipSystemStatsEntry 30 }

ipSystemStatsHCOutTransmits OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of IP datagrams that this entity supplied to the lower layers for transmission. This object counts the same datagrams as ipSystemStatsOutTransmits, but allows for larger values.

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

::= { ipSystemStatsEntry 31 }

ipSystemStatsOutOctets OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of octets in IP datagrams delivered to the lower layers for transmission. Octets from datagrams counted in ipSystemStatsOutTransmits MUST be counted here.

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

::= { ipSystemStatsEntry 32 }

ipSystemStatsHCOutOctets OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of octets in IP datagrams delivered to the lower layers for transmission. This object counts the same octets as ipSystemStatsOutOctets, but allows for larger values.

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

```
        ipSystemStatsDiscontinuityTime."  
 ::= { ipSystemStatsEntry 33 }
```

ipSystemStatsInMcastPkts OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of IP multicast datagrams received.

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

```
        ipSystemStatsDiscontinuityTime."  
 ::= { ipSystemStatsEntry 34 }
```

ipSystemStatsHCInMcastPkts OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of IP multicast datagrams received. This object counts the same datagrams as ipSystemStatsInMcastPkts but allows for larger values.

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

```
        ipSystemStatsDiscontinuityTime."  
 ::= { ipSystemStatsEntry 35 }
```

ipSystemStatsInMcastOctets OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of octets received in IP multicast datagrams. Octets from datagrams counted in ipSystemStatsInMcastPkts MUST be counted here.

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

```
        ipSystemStatsDiscontinuityTime."  
 ::= { ipSystemStatsEntry 36 }
```

ipSystemStatsHCInMcastOctets OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only
STATUS current
DESCRIPTION

"The total number of octets received in IP multicast datagrams. This object counts the same octets as ipSystemStatsInMcastOctets, but allows for larger values.

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

::= { ipSystemStatsEntry 37 }

ipSystemStatsOutMcastPkts OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"The number of IP multicast datagrams transmitted.

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

::= { ipSystemStatsEntry 38 }

ipSystemStatsHCOutMcastPkts OBJECT-TYPE

SYNTAX Counter64
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"The number of IP multicast datagrams transmitted. This object counts the same datagrams as ipSystemStatsOutMcastPkts, but allows for larger values.

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

::= { ipSystemStatsEntry 39 }

ipSystemStatsOutMcastOctets OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"The total number of octets transmitted in IP multicast datagrams. Octets from datagrams counted in

ipSystemStatsOutMcastPkts MUST be counted here.

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

::= { ipSystemStatsEntry 40 }

ipSystemStatsHCOutMcastOctets OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of octets transmitted in IP multicast datagrams. This object counts the same octets as ipSystemStatsOutMcastOctets, but allows for larger values.

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

::= { ipSystemStatsEntry 41 }

ipSystemStatsInBcastPkts OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of IP broadcast datagrams received.

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

::= { ipSystemStatsEntry 42 }

ipSystemStatsHCInBcastPkts OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of IP broadcast datagrams received. This object counts the same datagrams as ipSystemStatsInBcastPkts but allows for larger values.

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

```
        ipSystemStatsDiscontinuityTime."  
 ::= { ipSystemStatsEntry 43 }
```

ipSystemStatsOutBcastPkts OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of IP broadcast datagrams transmitted.

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

```
        ipSystemStatsDiscontinuityTime."  
 ::= { ipSystemStatsEntry 44 }
```

ipSystemStatsHCOutBcastPkts OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of IP broadcast datagrams transmitted. This object counts the same datagrams as ipSystemStatsOutBcastPkts, but allows for larger values.

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

```
        ipSystemStatsDiscontinuityTime."  
 ::= { ipSystemStatsEntry 45 }
```

ipSystemStatsDiscontinuityTime OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of sysUpTime on the most recent occasion at which any one or more of this entry's counters suffered a discontinuity.

If no such discontinuities have occurred since the last re-initialization of the local management subsystem, then this object contains a zero value."

```
 ::= { ipSystemStatsEntry 46 }
```

ipSystemStatsRefreshRate OBJECT-TYPE

SYNTAX Unsigned32

UNITS "milli-seconds"

MAX-ACCESS read-only
 STATUS current
 DESCRIPTION

"The minimum reasonable polling interval for this entry.

This object provides an indication of the minimum amount of time required to update the counters in this entry."

::= { ipSystemStatsEntry 47 }

ipIfStatsTableLastChange OBJECT-TYPE

SYNTAX TimeStamp
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION

"The value of sysUpTime on the most recent occasion at which a row in the ipIfStatsTable was added or deleted.

If new objects are added to the ipIfStatsTable that require the ipIfStatsTableLastChange to be updated when they are modified, they must specify that requirement in their description clause."

::= { ipTrafficStats 2 }

ipIfStatsTable OBJECT-TYPE

SYNTAX SEQUENCE OF IpIfStatsEntry
 MAX-ACCESS not-accessible
 STATUS current
 DESCRIPTION

"The table containing per-interface traffic statistics. This table and the ipSystemStatsTable contain similar objects whose difference is in their granularity. Where this table contains per-interface statistics, the ipSystemStatsTable contains the same statistics, but counted on a system wide basis."

::= { ipTrafficStats 3 }

ipIfStatsEntry OBJECT-TYPE

SYNTAX IpIfStatsEntry
 MAX-ACCESS not-accessible
 STATUS current
 DESCRIPTION

"An interface statistics entry containing objects for a particular interface and version of IP."

INDEX { ipIfStatsIPVersion, ipIfStatsIfIndex }

::= { ipIfStatsTable 1 }

IpIfStatsEntry ::= SEQUENCE {

ipIfStatsIPVersion	InetVersion,
ipIfStatsIfIndex	InterfaceIndex,

```

    ipIfStatsInReceives          Counter32,
    ipIfStatsHCInReceives        Counter64,
    ipIfStatsInOctets            Counter32,
    ipIfStatsHCInOctets          Counter64,
    ipIfStatsInHdrErrors         Counter32,
    ipIfStatsInNoRoutes          Counter32,
    ipIfStatsInAddrErrors        Counter32,
    ipIfStatsInUnknownProtos     Counter32,
    ipIfStatsInTruncatedPkts     Counter32,
    ipIfStatsInForwDatagrams     Counter32,
    ipIfStatsHCInForwDatagrams   Counter64,
    ipIfStatsReasmReqds          Counter32,
    ipIfStatsReasmOKs            Counter32,
    ipIfStatsReasmFails          Counter32,
    ipIfStatsInDiscards          Counter32,
    ipIfStatsInDelivers          Counter32,
    ipIfStatsHCInDelivers        Counter64,
    ipIfStatsOutRequests         Counter32,
    ipIfStatsHCOutRequests       Counter64,
    ipIfStatsOutForwDatagrams    Counter32,
    ipIfStatsHCOutForwDatagrams  Counter64,
    ipIfStatsOutDiscards         Counter32,
    ipIfStatsOutFragReqds        Counter32,
    ipIfStatsOutFragOKs          Counter32,
    ipIfStatsOutFragFails        Counter32,
    ipIfStatsOutFragCreates       Counter32,
    ipIfStatsOutTransmits         Counter32,
    ipIfStatsHCOutTransmits       Counter64,
    ipIfStatsOutOctets           Counter32,
    ipIfStatsHCOutOctets          Counter64,
    ipIfStatsInMcastPkts         Counter32,
    ipIfStatsHCInMcastPkts       Counter64,
    ipIfStatsInMcastOctets       Counter32,
    ipIfStatsHCInMcastOctets     Counter64,
    ipIfStatsOutMcastPkts        Counter32,
    ipIfStatsHCOutMcastPkts      Counter64,
    ipIfStatsOutMcastOctets      Counter32,
    ipIfStatsHCOutMcastOctets    Counter64,
    ipIfStatsInBcastPkts         Counter32,
    ipIfStatsHCInBcastPkts       Counter64,
    ipIfStatsOutBcastPkts        Counter32,
    ipIfStatsHCOutBcastPkts      Counter64,
    ipIfStatsDiscontinuityTime   TimeStamp,
    ipIfStatsRefreshRate         Unsigned32
}

```

```

ipIfStatsIPVersion OBJECT-TYPE
    SYNTAX      InetVersion

```

MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
 "The IP version of this row."
 ::= { ipIfStatsEntry 1 }

ipIfStatsIfIndex OBJECT-TYPE
SYNTAX InterfaceIndex
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
 "The index value that uniquely identifies the interface to
 which this entry is applicable. The interface identified by
 a particular value of this index is the same interface as
 identified by the same value of the IF-MIB's ifIndex."
 ::= { ipIfStatsEntry 2 }

ipIfStatsInReceives OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "The total number of input IP datagrams received, including
 those received in error.

 Discontinuities in the value of this counter can occur at
 re-initialization of the management system, and at other
 times as indicated by the value of
 ipIfStatsDiscontinuityTime."
 ::= { ipIfStatsEntry 3 }

ipIfStatsHCInReceives OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "The total number of input IP datagrams received, including
 those received in error. This object counts the same
 datagrams as ipIfStatsInReceives, but allows for larger
 values.

 Discontinuities in the value of this counter can occur at
 re-initialization of the management system, and at other
 times as indicated by the value of
 ipIfStatsDiscontinuityTime."
 ::= { ipIfStatsEntry 4 }

ipIfStatsInOctets OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"The total number of octets received in input IP datagrams, including those received in error. Octets from datagrams counted in ipIfStatsInReceives MUST be counted here.

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

::= { ipIfStatsEntry 5 }

ipIfStatsHCInOctets OBJECT-TYPE

SYNTAX Counter64
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"The total number of octets received in input IP datagrams, including those received in error. This object counts the same octets as ipIfStatsInOctets, but allows for larger values.

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

::= { ipIfStatsEntry 6 }

ipIfStatsInHdrErrors OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"The number of input IP datagrams discarded due to errors in their IP headers, including version number mismatch, other format errors, hop count exceeded, errors discovered in processing their IP options, etc.

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

::= { ipIfStatsEntry 7 }

ipIfStatsInNoRoutes OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only
STATUS current
DESCRIPTION

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

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

::= { ipIfStatsEntry 8 }

ipIfStatsInAddrErrors OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION

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

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

::= { ipIfStatsEntry 9 }

ipIfStatsInUnknownProtos OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"The number of locally-addressed IP datagrams received successfully but discarded because of an unknown or unsupported protocol.

When tracking interface statistics, the counter of the interface to which these datagrams were addressed is incremented. This interface might not be the same as the input interface for some of the datagrams.

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

```
        ipIfStatsDiscontinuityTime."  
 ::= { ipIfStatsEntry 10 }
```

ipIfStatsInTruncatedPkts OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

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

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

```
        ipIfStatsDiscontinuityTime."  
 ::= { ipIfStatsEntry 11 }
```

ipIfStatsInForwDatagrams OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of input datagrams for which this entity was not their final IP destination and for which this entity attempted to find a route to forward them to that final destination. In entities that do not act as IP routers, this counter will include only those datagrams that were Source-Routed via this entity, and the Source-Route processing was successful.

When tracking interface statistics, the counter of the incoming interface is incremented for each datagram.

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

```
        ipIfStatsDiscontinuityTime."  
 ::= { ipIfStatsEntry 12 }
```

ipIfStatsHCInForwDatagrams OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of input datagrams for which this entity was not their final IP destination and for which this entity attempted to find a route to forward them to that final destination. This object counts the same packets as

ipIfStatsInForwDatagrams, but allows for larger values.

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

::= { ipIfStatsEntry 13 }

ipIfStatsReasmReqds OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of IP fragments received that needed to be reassembled at this interface.

When tracking interface statistics, the counter of the interface to which these fragments were addressed is incremented. This interface might not be the same as the input interface for some of the fragments.

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

::= { ipIfStatsEntry 14 }

ipIfStatsReasmOKs OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of IP datagrams successfully reassembled.

When tracking interface statistics, the counter of the interface to which these datagrams were addressed is incremented. This interface might not be the same as the input interface for some of the datagrams.

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

::= { ipIfStatsEntry 15 }

ipIfStatsReasmFails OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of failures detected by the IP re-assembly algorithm (for whatever reason: timed out, errors, etc.). Note that this is not necessarily a count of discarded IP 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.

When tracking interface statistics, the counter of the interface to which these fragments were addressed is incremented. This interface might not be the same as the input interface for some of the fragments.

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

::= { ipIfStatsEntry 16 }

ipIfStatsInDiscards OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

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

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

::= { ipIfStatsEntry 17 }

ipIfStatsInDelivers OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of datagrams successfully delivered to IP user-protocols (including ICMP).

When tracking interface statistics, the counter of the interface to which these datagrams were addressed is incremented. This interface might not be the same as the

input interface for some of the datagrams.

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

::= { ipIfStatsEntry 18 }

ipIfStatsHCInDelivers OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of datagrams successfully delivered to IP user-protocols (including ICMP). This object counts the same packets as ipIfStatsInDelivers, but allows for larger values.

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

::= { ipIfStatsEntry 19 }

ipIfStatsOutRequests OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of IP datagrams that local IP user-protocols (including ICMP) supplied to IP in requests for transmission. Note that this counter does not include any datagrams counted in ipIfStatsOutForwDatagrams.

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

::= { ipIfStatsEntry 20 }

ipIfStatsHCOutRequests OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of IP datagrams that local IP user-protocols (including ICMP) supplied to IP in requests for transmission. This object counts the same packets as

ipIfStatsOutRequests, but allows for larger values.

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

::= { ipIfStatsEntry 21 }

-- This object ID is reserved to allow the IDs for this table's objects
-- to align with the objects in the ipSystemStatsTable.
-- ::= { ipIfStatsEntry 22 }

ipIfStatsOutForwDatagrams OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of datagrams for which this entity was not their final IP destination and for which it was successful in finding a path to their final destination. In entities that do not act as IP routers, this counter will include only those datagrams that were Source-Routed via this entity, and the Source-Route processing was successful.

When tracking interface statistics, the counter of the outgoing interface is incremented for a successfully forwarded datagram.

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

::= { ipIfStatsEntry 23 }

ipIfStatsHCOutForwDatagrams OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of datagrams for which this entity was not their final IP destination and for which it was successful in finding a path to their final destination. This object counts the same packets as ipIfStatsOutForwDatagrams, but allows for larger values.

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

```
        ipIfStatsDiscontinuityTime."  
 ::= { ipIfStatsEntry 24 }
```

ipIfStatsOutDiscards OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of output IP datagrams for which no problem was encountered to prevent their transmission to their destination, but were discarded (e.g., for lack of buffer space). Note that this counter would include datagrams counted in ipIfStatsOutForwDatagrams if any such datagrams met this (discretionary) discard criterion.

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

```
        ipIfStatsDiscontinuityTime."  
 ::= { ipIfStatsEntry 25 }
```

ipIfStatsOutFragReqds OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of IP datagrams that would require fragmentation in order to be transmitted.

When tracking interface statistics, the counter of the outgoing interface is incremented for a successfully fragmented datagram.

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

```
        ipIfStatsDiscontinuityTime."  
 ::= { ipIfStatsEntry 26 }
```

ipIfStatsOutFragOKs OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of IP datagrams that have been successfully fragmented.

When tracking interface statistics, the counter of the

outgoing interface is incremented for a successfully fragmented datagram.

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

::= { ipIfStatsEntry 27 }

ipIfStatsOutFragFails OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of IP datagrams that have been discarded because they needed to be fragmented but could not be. This includes IPv4 packets that have the DF bit set and IPv6 packets that are being forwarded and exceed the outgoing link MTU.

When tracking interface statistics, the counter of the outgoing interface is incremented for an unsuccessfully fragmented datagram.

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

::= { ipIfStatsEntry 28 }

ipIfStatsOutFragCreates 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 IP fragmentation.

When tracking interface statistics, the counter of the outgoing interface is incremented for a successfully fragmented datagram.

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

::= { ipIfStatsEntry 29 }

`ipIfStatsOutTransmits OBJECT-TYPE``SYNTAX Counter32``MAX-ACCESS read-only``STATUS current``DESCRIPTION`

"The total number of IP datagrams that this entity supplied to the lower layers for transmission. This includes datagrams generated locally and those forwarded by 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 the value of

`ipIfStatsDiscontinuityTime`."

::= { `ipIfStatsEntry` 30 }

`ipIfStatsHCOutTransmits OBJECT-TYPE``SYNTAX Counter64``MAX-ACCESS read-only``STATUS current``DESCRIPTION`

"The total number of IP datagrams that this entity supplied to the lower layers for transmission. This object counts the same datagrams as `ipIfStatsOutTransmits`, but allows for larger values.

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

`ipIfStatsDiscontinuityTime`."

::= { `ipIfStatsEntry` 31 }

`ipIfStatsOutOctets OBJECT-TYPE``SYNTAX Counter32``MAX-ACCESS read-only``STATUS current``DESCRIPTION`

"The total number of octets in IP datagrams delivered to the lower layers for transmission. Octets from datagrams counted in `ipIfStatsOutTransmits` MUST be counted here.

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

`ipIfStatsDiscontinuityTime`."

::= { `ipIfStatsEntry` 32 }

`ipIfStatsHCOctets OBJECT-TYPE`

SYNTAX Counter64
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"The total number of octets in IP datagrams delivered to the lower layers for transmission. This object counts the same octets as ipIfStatsOutOctets, but allows for larger values.

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

::= { ipIfStatsEntry 33 }

ipIfStatsInMcastPkts OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"The number of IP multicast datagrams received.

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

::= { ipIfStatsEntry 34 }

ipIfStatsHCInMcastPkts OBJECT-TYPE

SYNTAX Counter64
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"The number of IP multicast datagrams received. This object counts the same datagrams as ipIfStatsInMcastPkts, but allows for larger values.

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

::= { ipIfStatsEntry 35 }

ipIfStatsInMcastOctets OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"The total number of octets received in IP multicast

datagrams. Octets from datagrams counted in
ipIfStatsInMcastPkts MUST be counted here.

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

::= { ipIfStatsEntry 36 }

ipIfStatsHCInMcastOctets OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of octets received in IP multicast
datagrams. This object counts the same octets as
ipIfStatsInMcastOctets, but allows for larger values.

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

::= { ipIfStatsEntry 37 }

ipIfStatsOutMcastPkts OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of IP multicast datagrams transmitted.

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

::= { ipIfStatsEntry 38 }

ipIfStatsHCOutMcastPkts OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of IP multicast datagrams transmitted. This
object counts the same datagrams as ipIfStatsOutMcastPkts,
but allows for larger values.

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

times as indicated by the value of
ipIfStatsDiscontinuityTime."
 ::= { ipIfStatsEntry 39 }

ipIfStatsOutMcastOctets OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of octets transmitted in IP multicast datagrams. Octets from datagrams counted in ipIfStatsOutMcastPkts MUST be counted here.

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

::= { ipIfStatsEntry 40 }

ipIfStatsHCOutMcastOctets OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of octets transmitted in IP multicast datagrams. This object counts the same octets as ipIfStatsOutMcastOctets, but allows for larger values.

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

::= { ipIfStatsEntry 41 }

ipIfStatsInBcastPkts OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of IP broadcast datagrams received.

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

::= { ipIfStatsEntry 42 }

ipIfStatsHCInBcastPkts OBJECT-TYPE

SYNTAX Counter64
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"The number of IP broadcast datagrams received. This object counts the same datagrams as ipIfStatsInBcastPkts, but allows for larger values.

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

::= { ipIfStatsEntry 43 }

ipIfStatsOutBcastPkts OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"The number of IP broadcast datagrams transmitted.

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

::= { ipIfStatsEntry 44 }

ipIfStatsHCOutBcastPkts OBJECT-TYPE

SYNTAX Counter64
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"The number of IP broadcast datagrams transmitted. This object counts the same datagrams as ipIfStatsOutBcastPkts, but allows for larger values.

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

::= { ipIfStatsEntry 45 }

ipIfStatsDiscontinuityTime OBJECT-TYPE

SYNTAX TimeStamp
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"The value of sysUpTime on the most recent occasion at which

any one or more of this entry's counters suffered a discontinuity.

If no such discontinuities have occurred since the last re-initialization of the local management subsystem, then this object contains a zero value."

::= { ipIfStatsEntry 46 }

ipIfStatsRefreshRate OBJECT-TYPE

SYNTAX Unsigned32

UNITS "milli-seconds"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The minimum reasonable polling interval for this entry.

This object provides an indication of the minimum amount of time required to update the counters in this entry."

::= { ipIfStatsEntry 47 }

--

-- Internet Address Prefix table

--

ipAddressPrefixTable OBJECT-TYPE

SYNTAX SEQUENCE OF IpAddressPrefixEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table allows the user to determine the source of an IP address or set of IP addresses, and allows other tables to share the information via pointer rather than by copying.

For example, when the node configures both a unicast and anycast address for a prefix, the ipAddressPrefix objects for those addresses will point to a single row in this table.

This table primarily provides support for IPv6 prefixes, and several of the objects are less meaningful for IPv4. The table continues to allow IPv4 addresses to allow future flexibility. In order to promote a common configuration, this document includes suggestions for default values for IPv4 prefixes. Each of these values may be overridden if an object is meaningful to the node.

All prefixes used by this entity should be included in this table independent of how the entity learned the prefix.
(This table isn't limited to prefixes learned from router

```

        advertisements.)"
 ::= { ip 32 }

```

ipAddressPrefixEntry OBJECT-TYPE

```

SYNTAX      IPAddressPrefixEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "An entry in the ipAddressPrefixTable."
INDEX       { ipAddressPrefixIfIndex, ipAddressPrefixType,
              ipAddressPrefixPrefix, ipAddressPrefixLength }
 ::= { ipAddressPrefixTable 1 }

```

IpAddressPrefixEntry ::= SEQUENCE {

ipAddressPrefixIfIndex	InterfaceIndex,
ipAddressPrefixType	InetAddressType,
ipAddressPrefixPrefix	InetAddress,
ipAddressPrefixLength	InetAddressPrefixLength,
ipAddressPrefixOrigin	IpAddressPrefixOriginTC,
ipAddressPrefixOnLinkFlag	TruthValue,
ipAddressPrefixAutonomousFlag	TruthValue,
ipAddressPrefixAdvPreferredLifetime	Unsigned32,
ipAddressPrefixAdvValidLifetime	Unsigned32

```

}

```

ipAddressPrefixIfIndex OBJECT-TYPE

```

SYNTAX      InterfaceIndex
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "The index value that uniquely identifies the interface on
     which this prefix is configured. The interface identified
     by a particular value of this index is the same interface as
     identified by the same value of the IF-MIB's ifIndex."
 ::= { ipAddressPrefixEntry 1 }

```

ipAddressPrefixType OBJECT-TYPE

```

SYNTAX      InetAddressType
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "The address type of ipAddressPrefix."
 ::= { ipAddressPrefixEntry 2 }

```

ipAddressPrefixPrefix OBJECT-TYPE

```

SYNTAX      InetAddress
MAX-ACCESS  not-accessible
STATUS      current

```

DESCRIPTION

"The address prefix. The address type of this object is specified in ipAddressPrefixType. The length of this object is the standard length for objects of that type (4 or 16 bytes). Any bits after ipAddressPrefixLength must be zero.

Implementors need to be aware that, if the size of ipAddressPrefixPrefix exceeds 114 octets, then OIDs of instances of columns in this row will have more than 128 sub-identifiers and cannot be accessed using SNMPv1, SNMPv2c, or SNMPv3."

::= { ipAddressPrefixEntry 3 }

ipAddressPrefixLength OBJECT-TYPE

SYNTAX InetAddressPrefixLength

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The prefix length associated with this prefix.

The value 0 has no special meaning for this object. It simply refers to address '::/0'."

::= { ipAddressPrefixEntry 4 }

ipAddressPrefixOrigin OBJECT-TYPE

SYNTAX IpAddressPrefixOriginTC

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The origin of this prefix."

::= { ipAddressPrefixEntry 5 }

ipAddressPrefixOnLinkFlag 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; otherwise, the value is 'false(2)'."

The default for IPv4 prefixes is 'true(1)'."

REFERENCE "For IPv6 RFC 2461, especially sections 2 and 4.6.2 and RFC 2462"

::= { ipAddressPrefixEntry 6 }

ipAddressPrefixAutonomousFlag 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 auto-configure a local interface address.

The default for IPv4 prefixes is 'false(2)'."

REFERENCE "For IPv6 RFC 2461, especially sections 2 and 4.6.2 and RFC 2462"

::= { ipAddressPrefixEntry 7 }

ipAddressPrefixAdvPreferredLifetime OBJECT-TYPE

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

"The remaining length of time, in seconds, that this prefix will continue to be 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.

The default for IPv4 prefixes is 4,294,967,295 (infinity)."

REFERENCE "For IPv6 RFC 2461, especially sections 2 and 4.6.2 and RFC 2462"

::= { ipAddressPrefixEntry 8 }

ipAddressPrefixAdvValidLifetime OBJECT-TYPE

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

"The remaining length of time, in seconds, that this prefix will continue to be 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.

The default for IPv4 prefixes is 4,294,967,295 (infinity)."
REFERENCE "For IPv6 RFC 2461, especially sections 2 and 4.6.2 and
RFC 2462"

::= { ipAddressPrefixEntry 9 }

--
-- Internet Address Table
--

ipAddressSpinLock OBJECT-TYPE

SYNTAX TestAndIncr

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"An advisory lock used to allow cooperating SNMP managers to coordinate their use of the set operation in creating or modifying rows within this table.

In order to use this lock to coordinate the use of set operations, managers should first retrieve ipAddressTableSpinLock. They should then determine the appropriate row to create or modify. Finally, they should issue the appropriate set command, including the retrieved value of ipAddressSpinLock. If another manager has altered the table in the meantime, then the value of ipAddressSpinLock will have changed, and the creation will fail as it will be specifying an incorrect value for ipAddressSpinLock. It is suggested, but not required, that the ipAddressSpinLock be the first var bind for each set of objects representing a 'row' in a PDU."

::= { ip 33 }

ipAddressTable OBJECT-TYPE

SYNTAX SEQUENCE OF IpAddressEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

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

This table does not contain multicast address information. Tables for such information should be contained in multicast specific MIBs, such as RFC 3019.

While this table is writable, the user will note that several objects, such as ipAddressOrigin, are not. The intention in allowing a user to write to this table is to allow them to add or remove any entry that isn't

permanent. The user should be allowed to modify objects and entries when that would not cause inconsistencies within the table. Allowing write access to objects, such as ipAddressOrigin, could allow a user to insert an entry and then label it incorrectly.

Note well: When including IPv6 link-local addresses in this table, the entry must use an InetAddressType of 'ipv6z' in order to differentiate between the possible interfaces."

```
::= { ip 34 }
```

```
ipAddressEntry OBJECT-TYPE
    SYNTAX      InetAddressEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An address mapping for a particular interface."
    INDEX { ipAddressAddrType, ipAddressAddr }
    ::= { ipAddressTable 1 }
```

```
IpAddressEntry ::= SEQUENCE {
    ipAddressAddrType      InetAddressType,
    ipAddressAddr          InetAddress,
    ipAddressIfIndex       InterfaceIndex,
    ipAddressType          INTEGER,
    ipAddressPrefix        RowPointer,
    ipAddressOrigin        IpAddressOriginTC,
    ipAddressStatus        IpAddressStatusTC,
    ipAddressCreated       TimeStamp,
    ipAddressLastChanged   TimeStamp,
    ipAddressRowStatus     RowStatus,
    ipAddressStorageType   StorageType
}
```

```
ipAddressAddrType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The address type of ipAddressAddr."
    ::= { ipAddressEntry 1 }
```

```
ipAddressAddr OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The IP address to which this entry's addressing information
```

pertains. The address type of this object is specified in `ipAddressAddrType`.

Implementors need to be aware that if the size of `ipAddressAddr` exceeds 116 octets, then OIDS of instances of columns in this row will have more than 128 sub-identifiers and cannot be accessed using SNMPv1, SNMPv2c, or SNMPv3."

::= { `ipAddressEntry` 2 }

`ipAddressIfIndex` OBJECT-TYPE

SYNTAX InterfaceIndex

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The index value that uniquely identifies the interface to which this entry is applicable. The interface identified by a particular value of this index is the same interface as identified by the same value of the IF-MIB's `ifIndex`."

::= { `ipAddressEntry` 3 }

`ipAddressType` OBJECT-TYPE

SYNTAX INTEGER {
 unicast(1),
 anycast(2),
 broadcast(3)
}

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The type of address. `broadcast(3)` is not a valid value for IPv6 addresses (RFC 3513)."

DEFVAL { unicast }

::= { `ipAddressEntry` 4 }

`ipAddressPrefix` OBJECT-TYPE

SYNTAX RowPointer

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"A pointer to the row in the prefix table to which this address belongs. May be { 0 0 } if there is no such row."

DEFVAL { zeroDotZero }

::= { `ipAddressEntry` 5 }

`ipAddressOrigin` OBJECT-TYPE

SYNTAX IpAddressOriginTC

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The origin of the address."

::= { ipAddressEntry 6 }

ipAddressStatus OBJECT-TYPE

SYNTAX IpAddressStatusTC

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The status of the address, describing if the address can be used for communication.

In the absence of other information, an IPv4 address is always preferred(1)."

DEFVAL { preferred }

::= { ipAddressEntry 7 }

ipAddressCreated OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of sysUpTime at the time this entry was created. If this entry was created prior to the last re-initialization of the local network management subsystem, then this object contains a zero value."

::= { ipAddressEntry 8 }

ipAddressLastChanged 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."

::= { ipAddressEntry 9 }

ipAddressRowStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The status of this conceptual row.

The RowStatus TC requires that this DESCRIPTION clause states under which circumstances other objects in this row

can be modified. The value of this object has no effect on whether other objects in this conceptual row can be modified.

A conceptual row can not be made active until the ipAddressIfIndex has been set to a valid index."

```
::= { ipAddressEntry 10 }
```

```
ipAddressStorageType OBJECT-TYPE
```

```
SYNTAX      StorageType
```

```
MAX-ACCESS read-create
```

```
STATUS      current
```

```
DESCRIPTION
```

"The storage type for this conceptual row. If this object has a value of 'permanent', then no other objects are required to be able to be modified."

```
DEFVAL { volatile }
```

```
::= { ipAddressEntry 11 }
```

```
--
```

```
-- the Internet Address Translation table
```

```
--
```

```
ipNetToPhysicalTable OBJECT-TYPE
```

```
SYNTAX      SEQUENCE OF IpNetToPhysicalEntry
```

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

```
STATUS      current
```

```
DESCRIPTION
```

"The IP Address Translation table used for mapping from IP addresses to physical addresses.

The Address Translation tables contain the IP address to 'physical' address equivalences. Some interfaces do not use translation tables for determining address equivalences (e.g., DDN-X.25 has an algorithmic method); if all interfaces are of this type, then the Address Translation table is empty, i.e., has zero entries.

While many protocols may be used to populate this table, ARP and Neighbor Discovery are the most likely options."

```
REFERENCE "RFC 826 and RFC 2461"
```

```
::= { ip 35 }
```

```
ipNetToPhysicalEntry OBJECT-TYPE
```

```
SYNTAX      IpNetToPhysicalEntry
```

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

```
STATUS      current
```

DESCRIPTION

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

```
INDEX      { ipNetToPhysicalIfIndex,
              ipNetToPhysicalNetAddressType,
              ipNetToPhysicalNetAddress }
 ::= { ipNetToPhysicalTable 1 }
```

```
IpNetToPhysicalEntry ::= SEQUENCE {
    ipNetToPhysicalIfIndex      InterfaceIndex,
    ipNetToPhysicalNetAddressType  InetAddressType,
    ipNetToPhysicalNetAddress    InetAddress,
    ipNetToPhysicalPhysAddress    PhysAddress,
    ipNetToPhysicalLastUpdated    TimeStamp,
    ipNetToPhysicalType          INTEGER,
    ipNetToPhysicalState          INTEGER,
    ipNetToPhysicalRowStatus      RowStatus
}
```

ipNetToPhysicalIfIndex OBJECT-TYPE

```
SYNTAX      InterfaceIndex
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
```

"The index value that uniquely identifies the interface to which this entry is applicable. The interface identified by a particular value of this index is the same interface as identified by the same value of the IF-MIB's ifIndex."

```
::= { ipNetToPhysicalEntry 1 }
```

ipNetToPhysicalNetAddressType OBJECT-TYPE

```
SYNTAX      InetAddressType
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
```

"The type of ipNetToPhysicalNetAddress."

```
::= { ipNetToPhysicalEntry 2 }
```

ipNetToPhysicalNetAddress OBJECT-TYPE

```
SYNTAX      InetAddress
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
```

"The IP Address corresponding to the media-dependent 'physical' address. The address type of this object is specified in ipNetToPhysicalAddressType."

Implementors need to be aware that if the size of

ipNetToPhysicalNetAddress exceeds 115 octets, then OIDS of instances of columns in this row will have more than 128 sub-identifiers and cannot be accessed using SNMPv1, SNMPv2c, or SNMPv3."

::= { ipNetToPhysicalEntry 3 }

ipNetToPhysicalPhysAddress OBJECT-TYPE

SYNTAX PhysAddress (SIZE(0..65535))

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The media-dependent 'physical' address.

As the entries in this table are typically not persistent when this object is written the entity SHOULD NOT save the change to non-volatile storage."

::= { ipNetToPhysicalEntry 4 }

ipNetToPhysicalLastUpdated 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."

::= { ipNetToPhysicalEntry 5 }

ipNetToPhysicalType OBJECT-TYPE

SYNTAX INTEGER {
 other(1), -- none of the following
 invalid(2), -- an invalidated mapping
 dynamic(3),
 static(4),
 local(5) -- local interface
 }

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The type of mapping.

Setting this object to the value invalid(2) has the effect of invalidating the corresponding entry in the ipNetToPhysicalTable. That is, it effectively dis-associates 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 ipNetToPhysicalType object.

The 'dynamic(3)' type indicates that the IP address to physical addresses mapping has been dynamically resolved using e.g., IPv4 ARP or the IPv6 Neighbor Discovery protocol.

The 'static(4)' type indicates that the mapping has been statically configured. Both of these refer to entries that provide mappings for other entities addresses.

The 'local(5)' type indicates that the mapping is provided for an entity's own interface address.

As the entries in this table are typically not persistent when this object is written the entity SHOULD NOT save the change to non-volatile storage."

```
DEFVAL { static }
 ::= { ipNetToPhysicalEntry 6 }
```

ipNetToPhysicalState 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.
    incomplete(7) -- address resolution is being
                  -- performed.
}
MAX-ACCESS read-only
STATUS      current
DESCRIPTION
```

"The Neighbor Unreachability Detection state for the interface when the address mapping in this entry is used. If Neighbor Unreachability Detection is not in use (e.g. for IPv4), this object is always unknown(6)."

REFERENCE "RFC 2461"

::= { ipNetToPhysicalEntry 7 }

ipNetToPhysicalRowStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The status of this conceptual row.

The RowStatus TC requires that this DESCRIPTION clause states under which circumstances other objects in this row can be modified. The value of this object has no effect on whether other objects in this conceptual row can be modified.

A conceptual row can not be made active until the ipNetToPhysicalPhysAddress object has been set.

Note that if the ipNetToPhysicalType is set to 'invalid', the managed node may delete the entry independent of the state of this object."

::= { ipNetToPhysicalEntry 8 }

--

-- The IPv6 Scope Zone Index Table.

--

ipv6ScopeZoneIndexTable OBJECT-TYPE

SYNTAX SEQUENCE OF Ipv6ScopeZoneIndexEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The table used to describe IPv6 unicast and multicast scope zones.

For those objects that have names rather than numbers, the names were chosen to coincide with the names used in the IPv6 address architecture document. "

REFERENCE "Section 2.7 of RFC 4291"

::= { ip 36 }

ipv6ScopeZoneIndexEntry OBJECT-TYPE

SYNTAX Ipv6ScopeZoneIndexEntry

MAX-ACCESS not-accessible
 STATUS current
 DESCRIPTION

"Each entry contains the list of scope identifiers on a given interface."

INDEX { ipv6ScopeZoneIndexIfIndex }
 ::= { ipv6ScopeZoneIndexTable 1 }

```

Ipv6ScopeZoneIndexEntry ::= SEQUENCE {
    ipv6ScopeZoneIndexIfIndex      InterfaceIndex,
    ipv6ScopeZoneIndexLinkLocal    InetZoneIndex,
    ipv6ScopeZoneIndex3            InetZoneIndex,
    ipv6ScopeZoneIndexAdminLocal   InetZoneIndex,
    ipv6ScopeZoneIndexSiteLocal    InetZoneIndex,
    ipv6ScopeZoneIndex6            InetZoneIndex,
    ipv6ScopeZoneIndex7            InetZoneIndex,
    ipv6ScopeZoneIndexOrganizationLocal InetZoneIndex,
    ipv6ScopeZoneIndex9            InetZoneIndex,
    ipv6ScopeZoneIndexA            InetZoneIndex,
    ipv6ScopeZoneIndexB            InetZoneIndex,
    ipv6ScopeZoneIndexC            InetZoneIndex,
    ipv6ScopeZoneIndexD            InetZoneIndex
}

```

ipv6ScopeZoneIndexIfIndex OBJECT-TYPE

SYNTAX InterfaceIndex
 MAX-ACCESS not-accessible
 STATUS current
 DESCRIPTION

"The index value that uniquely identifies the interface to which these scopes belong. The interface identified by a particular value of this index is the same interface as identified by the same value of the IF-MIB's ifIndex."

::= { ipv6ScopeZoneIndexEntry 1 }

ipv6ScopeZoneIndexLinkLocal OBJECT-TYPE

SYNTAX InetZoneIndex
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION

"The zone index for the link-local scope on this interface."

::= { ipv6ScopeZoneIndexEntry 2 }

ipv6ScopeZoneIndex3 OBJECT-TYPE

SYNTAX InetZoneIndex
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION

```
    "The zone index for scope 3 on this interface."  
 ::= { ipv6ScopeZoneIndexEntry 3 }
```

ipv6ScopeZoneIndexAdminLocal OBJECT-TYPE

SYNTAX InetZoneIndex

MAX-ACCESS read-only

STATUS current

DESCRIPTION

```
    "The zone index for the admin-local scope on this interface."  
 ::= { ipv6ScopeZoneIndexEntry 4 }
```

ipv6ScopeZoneIndexSiteLocal OBJECT-TYPE

SYNTAX InetZoneIndex

MAX-ACCESS read-only

STATUS current

DESCRIPTION

```
    "The zone index for the site-local scope on this interface."  
 ::= { ipv6ScopeZoneIndexEntry 5 }
```

ipv6ScopeZoneIndex6 OBJECT-TYPE

SYNTAX InetZoneIndex

MAX-ACCESS read-only

STATUS current

DESCRIPTION

```
    "The zone index for scope 6 on this interface."  
 ::= { ipv6ScopeZoneIndexEntry 6 }
```

ipv6ScopeZoneIndex7 OBJECT-TYPE

SYNTAX InetZoneIndex

MAX-ACCESS read-only

STATUS current

DESCRIPTION

```
    "The zone index for scope 7 on this interface."  
 ::= { ipv6ScopeZoneIndexEntry 7 }
```

ipv6ScopeZoneIndexOrganizationLocal OBJECT-TYPE

SYNTAX InetZoneIndex

MAX-ACCESS read-only

STATUS current

DESCRIPTION

```
    "The zone index for the organization-local scope on this  
    interface."  
 ::= { ipv6ScopeZoneIndexEntry 8 }
```

ipv6ScopeZoneIndex9 OBJECT-TYPE

SYNTAX InetZoneIndex

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The zone index for scope 9 on this interface."

::= { ipv6ScopeZoneIndexEntry 9 }

ipv6ScopeZoneIndexA OBJECT-TYPE

SYNTAX InetZoneIndex

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The zone index for scope A on this interface."

::= { ipv6ScopeZoneIndexEntry 10 }

ipv6ScopeZoneIndexB OBJECT-TYPE

SYNTAX InetZoneIndex

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The zone index for scope B on this interface."

::= { ipv6ScopeZoneIndexEntry 11 }

ipv6ScopeZoneIndexC OBJECT-TYPE

SYNTAX InetZoneIndex

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The zone index for scope C on this interface."

::= { ipv6ScopeZoneIndexEntry 12 }

ipv6ScopeZoneIndexD OBJECT-TYPE

SYNTAX InetZoneIndex

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The zone index for scope D on this interface."

::= { ipv6ScopeZoneIndexEntry 13 }

--

-- The Default Router Table

-- This table simply lists the default routers; for more information

-- about routing tables, see the routing MIBs

--

ipDefaultRouterTable OBJECT-TYPE

SYNTAX SEQUENCE OF IpDefaultRouterEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The table used to describe the default routers known to this

```

        entity."
 ::= { ip 37 }

```

ipDefaultRouterEntry OBJECT-TYPE

SYNTAX IpDefaultRouterEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Each entry contains information about a default router known to this entity."

INDEX {ipDefaultRouterAddressType, ipDefaultRouterAddress, ipDefaultRouterIfIndex}

```
 ::= { ipDefaultRouterTable 1 }
```

IpDefaultRouterEntry ::= SEQUENCE {

ipDefaultRouterAddressType InetAddressType,

ipDefaultRouterAddress InetAddress,

ipDefaultRouterIfIndex InterfaceIndex,

ipDefaultRouterLifetime Unsigned32,

ipDefaultRouterPreference INTEGER

```
 }
```

ipDefaultRouterAddressType OBJECT-TYPE

SYNTAX InetAddressType

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The address type for this row."

```
 ::= { ipDefaultRouterEntry 1 }
```

ipDefaultRouterAddress OBJECT-TYPE

SYNTAX InetAddress

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The IP address of the default router represented by this row. The address type of this object is specified in ipDefaultRouterAddressType."

Implementers need to be aware that if the size of ipDefaultRouterAddress exceeds 115 octets, then OIDS of instances of columns in this row will have more than 128 sub-identifiers and cannot be accessed using SNMPv1, SNMPv2c, or SNMPv3."

```
 ::= { ipDefaultRouterEntry 2 }
```

ipDefaultRouterIfIndex OBJECT-TYPE

SYNTAX InterfaceIndex

MAX-ACCESS not-accessible
STATUS current
DESCRIPTION

"The index value that uniquely identifies the interface by which the router can be reached. The interface identified by a particular value of this index is the same interface as identified by the same value of the IF-MIB's ifIndex."

::= { ipDefaultRouterEntry 3 }

ipDefaultRouterLifetime OBJECT-TYPE

SYNTAX Unsigned32 (0..65535)

UNITS "seconds"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The remaining length of time, in seconds, that this router will continue to be useful as a default router. A value of zero indicates that it is no longer useful as a default router. It is left to the implementer of the MIB as to whether a router with a lifetime of zero is removed from the list.

For IPv6, this value should be extracted from the router advertisement messages."

REFERENCE "For IPv6 RFC 2462 sections 4.2 and 6.3.4"

::= { ipDefaultRouterEntry 4 }

ipDefaultRouterPreference OBJECT-TYPE

SYNTAX INTEGER {
 reserved (-2),
 low (-1),
 medium (0),
 high (1)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"An indication of preference given to this router as a default router as described in the Default Router Preferences document. Treating the value as a 2 bit signed integer allows for simple arithmetic comparisons.

For IPv4 routers or IPv6 routers that are not using the updated router advertisement format, this object is set to medium (0)."

REFERENCE "RFC 4291, section 2.1"

::= { ipDefaultRouterEntry 5 }

```
--  
-- Configuration information for constructing router advertisements  
--
```

ipv6RouterAdvertSpinLock OBJECT-TYPE

SYNTAX TestAndIncr

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"An advisory lock used to allow cooperating SNMP managers to coordinate their use of the set operation in creating or modifying rows within this table.

In order to use this lock to coordinate the use of set operations, managers should first retrieve ipv6RouterAdvertSpinLock. They should then determine the appropriate row to create or modify. Finally, they should issue the appropriate set command including the retrieved value of ipv6RouterAdvertSpinLock. If another manager has altered the table in the meantime, then the value of ipv6RouterAdvertSpinLock will have changed and the creation will fail as it will be specifying an incorrect value for ipv6RouterAdvertSpinLock. It is suggested, but not required, that the ipv6RouterAdvertSpinLock be the first var bind for each set of objects representing a 'row' in a PDU."

::= { ip 38 }

ipv6RouterAdvertTable OBJECT-TYPE

SYNTAX SEQUENCE OF Ipv6RouterAdvertEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The table containing information used to construct router advertisements."

::= { ip 39 }

ipv6RouterAdvertEntry OBJECT-TYPE

SYNTAX Ipv6RouterAdvertEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry containing information used to construct router advertisements.

Information in this table is persistent, and when this object is written, the entity SHOULD save the change to non-volatile storage."

INDEX { ipv6RouterAdvertIfIndex }

```
::= { ipv6RouterAdvertTable 1 }
```

```
Ipv6RouterAdvertEntry ::= SEQUENCE {
    ipv6RouterAdvertIfIndex      InterfaceIndex,
    ipv6RouterAdvertSendAdverts  TruthValue,
    ipv6RouterAdvertMaxInterval  Unsigned32,
    ipv6RouterAdvertMinInterval  Unsigned32,
    ipv6RouterAdvertManagedFlag TruthValue,
    ipv6RouterAdvertOtherConfigFlag TruthValue,
    ipv6RouterAdvertLinkMTU      Unsigned32,
    ipv6RouterAdvertReachableTime Unsigned32,
    ipv6RouterAdvertRetransmitTime Unsigned32,
    ipv6RouterAdvertCurHopLimit  Unsigned32,
    ipv6RouterAdvertDefaultLifetime Unsigned32,
    ipv6RouterAdvertRowStatus     RowStatus
}
```

```
ipv6RouterAdvertIfIndex OBJECT-TYPE
```

```
SYNTAX      InterfaceIndex
```

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

```
STATUS      current
```

```
DESCRIPTION
```

"The index value that uniquely identifies the interface on which router advertisements constructed with this information will be transmitted. The interface identified by a particular value of this index is the same interface as identified by the same value of the IF-MIB's ifIndex."

```
::= { ipv6RouterAdvertEntry 1 }
```

```
ipv6RouterAdvertSendAdverts OBJECT-TYPE
```

```
SYNTAX      TruthValue
```

```
MAX-ACCESS  read-create
```

```
STATUS      current
```

```
DESCRIPTION
```

"A flag indicating whether the router sends periodic router advertisements and responds to router solicitations on this interface."

```
REFERENCE "RFC 2461 Section 6.2.1"
```

```
DEFVAL { false }
```

```
::= { ipv6RouterAdvertEntry 2 }
```

```
ipv6RouterAdvertMaxInterval OBJECT-TYPE
```

```
SYNTAX      Unsigned32 (4..1800)
```

```
UNITS       "seconds"
```

```
MAX-ACCESS  read-create
```

```
STATUS      current
```

```
DESCRIPTION
```

"The maximum time allowed between sending unsolicited router

```
        advertisements from this interface."
REFERENCE "RFC 2461 Section 6.2.1"
DEFVAL { 600 }
 ::= { ipv6RouterAdvertEntry 3 }

ipv6RouterAdvertMinInterval OBJECT-TYPE
SYNTAX      Unsigned32 (3..1350)
UNITS       "seconds"
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "The minimum time allowed between sending unsolicited router
    advertisements from this interface.

    The default is 0.33 * ipv6RouterAdvertMaxInterval, however,
    in the case of a low value for ipv6RouterAdvertMaxInterval,
    the minimum value for this object is restricted to 3."
REFERENCE "RFC 2461 Section 6.2.1"
 ::= { ipv6RouterAdvertEntry 4 }

ipv6RouterAdvertManagedFlag OBJECT-TYPE
SYNTAX      TruthValue
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "The true/false value to be placed into the 'managed address
    configuration' flag field in router advertisements sent from
    this interface."
REFERENCE "RFC 2461 Section 6.2.1"
DEFVAL { false }
 ::= { ipv6RouterAdvertEntry 5 }

ipv6RouterAdvertOtherConfigFlag OBJECT-TYPE
SYNTAX      TruthValue
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "The true/false value to be placed into the 'other stateful
    configuration' flag field in router advertisements sent from
    this interface."
REFERENCE "RFC 2461 Section 6.2.1"
DEFVAL { false }
 ::= { ipv6RouterAdvertEntry 6 }

ipv6RouterAdvertLinkMTU OBJECT-TYPE
SYNTAX      Unsigned32
MAX-ACCESS  read-create
STATUS      current
```

DESCRIPTION

"The value to be placed in MTU options sent by the router on this interface.

A value of zero indicates that no MTU options are sent."

REFERENCE "RFC 2461 Section 6.2.1"

DEFVAL { 0 }

::= { ipv6RouterAdvertEntry 7 }

ipv6RouterAdvertReachableTime OBJECT-TYPE

SYNTAX Unsigned32 (0..3600000)

UNITS "milliseconds"

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The value to be placed in the reachable time field in router advertisement messages sent from this interface.

A value of zero in the router advertisement indicates that the advertisement isn't specifying a value for reachable time."

REFERENCE "RFC 2461 Section 6.2.1"

DEFVAL { 0 }

::= { ipv6RouterAdvertEntry 8 }

ipv6RouterAdvertRetransmitTime OBJECT-TYPE

SYNTAX Unsigned32

UNITS "milliseconds"

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The value to be placed in the retransmit timer field in router advertisements sent from this interface.

A value of zero in the router advertisement indicates that the advertisement isn't specifying a value for retrans time."

REFERENCE "RFC 2461 Section 6.2.1"

DEFVAL { 0 }

::= { ipv6RouterAdvertEntry 9 }

ipv6RouterAdvertCurHopLimit OBJECT-TYPE

SYNTAX Unsigned32 (0..255)

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The default value to be placed in the current hop limit field in router advertisements sent from this interface.

The value should be set to the current diameter of the Internet.

A value of zero in the router advertisement indicates that the advertisement isn't specifying a value for curHopLimit.

The default should be set to the value specified in the IANA web pages (www.iana.org) at the time of implementation."

REFERENCE "RFC 2461 Section 6.2.1"

::= { ipv6RouterAdvertEntry 10 }

ipv6RouterAdvertDefaultLifetime OBJECT-TYPE

SYNTAX Unsigned32 (0|4..9000)

UNITS "seconds"

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The value to be placed in the router lifetime field of router advertisements sent from this interface. This value MUST be either 0 or between ipv6RouterAdvertMaxInterval and 9000 seconds.

A value of zero indicates that the router is not to be used as a default router.

The default is 3 * ipv6RouterAdvertMaxInterval."

REFERENCE "RFC 2461 Section 6.2.1"

::= { ipv6RouterAdvertEntry 11 }

ipv6RouterAdvertRowStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The status of this conceptual row.

As all objects in this conceptual row have default values, a row can be created and made active by setting this object appropriately.

The RowStatus TC requires that this DESCRIPTION clause states under which circumstances other objects in this row can be modified. The value of this object has no effect on whether other objects in this conceptual row can be modified."

::= { ipv6RouterAdvertEntry 12 }

--

```
-- ICMP section
--

icmp      OBJECT IDENTIFIER ::= { mib-2 5 }

--
-- ICMP non-message-specific counters
--

-- These object IDs are reserved, as they were used in earlier
-- versions of the MIB module.  In theory, OIDs are not assigned
-- until the specification is released as an RFC; however, as some
-- companies may have shipped code based on earlier versions of
-- the MIB, it seems best to reserve these OIDs.
-- ::= { icmp 27 }
-- ::= { icmp 28 }

icmpStatsTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF IcmpStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The table of generic system-wide ICMP counters."
    ::= { icmp 29 }

icmpStatsEntry OBJECT-TYPE
    SYNTAX      IcmpStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A conceptual row in the icmpStatsTable."
    INDEX       { icmpStatsIPVersion }
    ::= { icmpStatsTable 1 }

IcmpStatsEntry ::= SEQUENCE {
    icmpStatsIPVersion      InetVersion,
    icmpStatsInMsgs         Counter32,
    icmpStatsInErrors       Counter32,
    icmpStatsOutMsgs        Counter32,
    icmpStatsOutErrors      Counter32
}

icmpStatsIPVersion OBJECT-TYPE
    SYNTAX      InetVersion
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The IP version of the statistics."
```

```
 ::= { icmpStatsEntry 1 }

icmpStatsInMsgs OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The total number of ICMP messages that the entity received.
        Note that this counter includes all those counted by
        icmpStatsInErrors."
    ::= { icmpStatsEntry 2 }

icmpStatsInErrors OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of ICMP messages that the entity received but
        determined as having ICMP-specific errors (bad ICMP
        checksums, bad length, etc.)."
    ::= { icmpStatsEntry 3 }

icmpStatsOutMsgs OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The total number of ICMP messages that the entity attempted
        to send. Note that this counter includes all those counted
        by icmpStatsOutErrors."
    ::= { icmpStatsEntry 4 }

icmpStatsOutErrors OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of ICMP messages that this entity did not send
        due to problems discovered within ICMP, such as a lack of
        buffers. This value should not include errors discovered
        outside the ICMP layer, such as the inability of IP to route
        the resultant datagram. In some implementations, there may
        be no types of error that contribute to this counter's
        value."
    ::= { icmpStatsEntry 5 }

--
-- per-version, per-message type ICMP counters
```

--

icmpMsgStatsTable OBJECT-TYPE

SYNTAX SEQUENCE OF IcmpMsgStatsEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The table of system-wide per-version, per-message type ICMP counters."

::= { icmp 30 }

icmpMsgStatsEntry OBJECT-TYPE

SYNTAX IcmpMsgStatsEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A conceptual row in the icmpMsgStatsTable.

The system should track each ICMP type value, even if that ICMP type is not supported by the system. However, a given row need not be instantiated unless a message of that type has been processed, i.e., the row for icmpMsgStatsType=X MAY be instantiated before but MUST be instantiated after the first message with Type=X is received or transmitted. After receiving or transmitting any succeeding messages with Type=X, the relevant counter must be incremented."

INDEX { icmpMsgStatsIPVersion, icmpMsgStatsType }

::= { icmpMsgStatsTable 1 }

IcmpMsgStatsEntry ::= SEQUENCE {

icmpMsgStatsIPVersion InetVersion,

icmpMsgStatsType Integer32,

icmpMsgStatsInPkts Counter32,

icmpMsgStatsOutPkts Counter32

}

icmpMsgStatsIPVersion OBJECT-TYPE

SYNTAX InetVersion

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The IP version of the statistics."

::= { icmpMsgStatsEntry 1 }

icmpMsgStatsType OBJECT-TYPE

SYNTAX Integer32 (0..255)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The ICMP type field of the message type being counted by this row.

Note that ICMP message types are scoped by the address type in use."

REFERENCE "http://www.iana.org/assignments/icmp-parameters and
http://www.iana.org/assignments/icmpv6-parameters"

::= { icmpMsgStatsEntry 2 }

icmpMsgStatsInPkts OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of input packets for this AF and type."

::= { icmpMsgStatsEntry 3 }

icmpMsgStatsOutPkts OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of output packets for this AF and type."

::= { icmpMsgStatsEntry 4 }

--

-- conformance information

--

ipMIBConformance OBJECT IDENTIFIER ::= { ipMIB 2 }

ipMIBCompliances OBJECT IDENTIFIER ::= { ipMIBConformance 1 }

ipMIBGroups OBJECT IDENTIFIER ::= { ipMIBConformance 2 }

-- compliance statements

ipMIBCompliance2 MODULE-COMPLIANCE

STATUS current

DESCRIPTION

"The compliance statement for systems that implement IP -
either IPv4 or IPv6.

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          ipSystemStatsIPVersion
-- SYNTAX          InetVersion {ipv4(1), ipv6(2)}
-- DESCRIPTION
--     This MIB requires support for only IPv4 and IPv6
--     versions.
--
-- OBJECT          ipIfStatsIPVersion
-- SYNTAX          InetVersion {ipv4(1), ipv6(2)}
-- DESCRIPTION
--     This MIB requires support for only IPv4 and IPv6
--     versions.
--
-- OBJECT          icmpStatsIPVersion
-- SYNTAX          InetVersion {ipv4(1), ipv6(2)}
-- DESCRIPTION
--     This MIB requires support for only IPv4 and IPv6
--     versions.
--
-- OBJECT          icmpMsgStatsIPVersion
-- SYNTAX          InetVersion {ipv4(1), ipv6(2)}
-- DESCRIPTION
--     This MIB requires support for only IPv4 and IPv6
--     versions.
--
-- OBJECT          ipAddressPrefixType
-- SYNTAX          InetAddressType {ipv4(1), ipv6(2)}
-- DESCRIPTION
--     This MIB requires support for only global IPv4 and
--     IPv6 address types.
--
-- OBJECT          ipAddressPrefixPrefix
-- SYNTAX          InetAddress (Size(4 | 16))
-- DESCRIPTION
--     This MIB requires support for only global IPv4 and
--     IPv6 addresses and so the size can be either 4 or
--     16 bytes.
--
-- OBJECT          ipAddressAddrType
-- SYNTAX          InetAddressType {ipv4(1), ipv6(2),
--                                  ipv4z(3), ipv6z(4)}
-- DESCRIPTION
--     This MIB requires support for only global and
--     non-global IPv4 and IPv6 address types.
--
-- OBJECT          ipAddressAddr
-- SYNTAX          InetAddress (Size(4 | 8 | 16 | 20))
-- DESCRIPTION
--     This MIB requires support for only global and
```

```

--      non-global IPv4 and IPv6 addresses and so the size
--      can be 4, 8, 16, or 20 bytes.
--
-- OBJECT          ipNetToPhysicalNetAddressType
-- SYNTAX          InetAddressType {ipv4(1), ipv6(2),
--                                ipv4z(3), ipv6z(4)}
-- DESCRIPTION
--      This MIB requires support for only global and
--      non-global IPv4 and IPv6 address types.
--
-- OBJECT          ipNetToPhysicalNetAddress
-- SYNTAX          InetAddress (Size(4 | 8 | 16 | 20))
-- DESCRIPTION
--      This MIB requires support for only global and
--      non-global IPv4 and IPv6 addresses and so the size
--      can be 4, 8, 16, or 20 bytes.
--
-- OBJECT          ipDefaultRouterAddressType
-- SYNTAX          InetAddressType {ipv4(1), ipv6(2),
--                                ipv4z(3), ipv6z(4)}
-- DESCRIPTION
--      This MIB requires support for only global and
--      non-global IPv4 and IPv6 address types.
--
-- OBJECT          ipDefaultRouterAddress
-- SYNTAX          InetAddress (Size(4 | 8 | 16 | 20))
-- DESCRIPTION
--      This MIB requires support for only global and
--      non-global IPv4 and IPv6 addresses and so the size
--      can be 4, 8, 16, or 20 bytes."

```

MODULE -- this module

```

MANDATORY-GROUPS { ipSystemStatsGroup,  ipAddressGroup,
                   ipNetToPhysicalGroup, ipDefaultRouterGroup,
                   icmpStatsGroup }

```

GROUP ipSystemStatsHCOctetGroup

DESCRIPTION

"This group is mandatory for systems that have an aggregate bandwidth of greater than 20MB. Including this group does not allow an entity to neglect the 32 bit versions of these objects."

GROUP ipSystemStatsHCPacketGroup

DESCRIPTION

"This group is mandatory for systems that have an aggregate bandwidth of greater than 650MB. Including this group

does not allow an entity to neglect the 32 bit versions of these objects."

GROUP ipIfStatsGroup

DESCRIPTION

"This group is optional for all systems."

GROUP ipIfStatsHCOctetGroup

DESCRIPTION

"This group is mandatory for systems that include the ipIfStatsGroup and include links with bandwidths of greater than 20MB. Including this group does not allow an entity to neglect the 32 bit versions of these objects."

GROUP ipIfStatsHCPacketGroup

DESCRIPTION

"This group is mandatory for systems that include the ipIfStatsGroup and include links with bandwidths of greater than 650MB. Including this group does not allow an entity to neglect the 32 bit versions of these objects."

GROUP ipv4GeneralGroup

DESCRIPTION

"This group is mandatory for all systems supporting IPv4."

GROUP ipv4IfGroup

DESCRIPTION

"This group is mandatory for all systems supporting IPv4."

GROUP ipv4SystemStatsGroup

DESCRIPTION

"This group is mandatory for all systems supporting IPv4."

GROUP ipv4SystemStatsHCPacketGroup

DESCRIPTION

"This group is mandatory for all systems supporting IPv4 and that have an aggregate bandwidth of greater than 650MB. Including this group does not allow an entity to neglect the 32 bit versions of these objects."

GROUP ipv4IfStatsGroup

DESCRIPTION

"This group is mandatory for all systems supporting IPv4 and including the ipIfStatsGroup."

GROUP ipv4IfStatsHCPacketGroup

DESCRIPTION

"This group is mandatory for all systems supporting IPv4 and

including the ipIfStatsHCPCPacketGroup. Including this group does not allow an entity to neglect the 32 bit versions of these objects."

GROUP ipv6GeneralGroup2

DESCRIPTION

"This group is mandatory for all systems supporting IPv6."

GROUP ipv6IfGroup

DESCRIPTION

"This group is mandatory for all systems supporting IPv6."

GROUP ipAddressPrefixGroup

DESCRIPTION

"This group is mandatory for all systems supporting IPv6."

GROUP ipv6ScopeGroup

DESCRIPTION

"This group is mandatory for all systems supporting IPv6."

GROUP ipv6RouterAdvertGroup

DESCRIPTION

"This group is mandatory for all IPv6 routers."

GROUP ipLastChangeGroup

DESCRIPTION

"This group is optional for all agents."

OBJECT ipv6IpForwarding

MIN-ACCESS read-only

DESCRIPTION

"An agent is not required to provide write access to this object."

OBJECT ipv6IpDefaultHopLimit

MIN-ACCESS read-only

DESCRIPTION

"An agent is not required to provide write access to this object."

OBJECT ipv4InterfaceEnableStatus

MIN-ACCESS read-only

DESCRIPTION

"An agent is not required to provide write access to this object."

OBJECT ipv6InterfaceEnableStatus

MIN-ACCESS read-only

DESCRIPTION

"An agent is not required to provide write access to this object."

OBJECT ipv6InterfaceForwarding

MIN-ACCESS read-only

DESCRIPTION

"An agent is not required to provide write access to this object."

OBJECT ipAddressSpinLock

MIN-ACCESS not-accessible

DESCRIPTION

"An agent is not required to provide write access to this object. However, if an agent provides write access to any of the other objects in the ipAddressGroup, it SHOULD provide write access to this object as well."

OBJECT ipAddressIfIndex

MIN-ACCESS read-only

DESCRIPTION

"An agent is not required to provide write or create access to this object."

OBJECT ipAddressType

MIN-ACCESS read-only

DESCRIPTION

"An agent is not required to provide write or create access to this object."

OBJECT ipAddressStatus

MIN-ACCESS read-only

DESCRIPTION

"An agent is not required to provide write or create access to this object."

OBJECT ipAddressRowStatus

SYNTAX RowStatus { active(1) }

MIN-ACCESS read-only

DESCRIPTION

"An agent is not required to provide write or create access to this object."

OBJECT ipAddressStorageType

MIN-ACCESS read-only

DESCRIPTION

"An agent is not required to provide write or create access to this object."

If an agent allows this object to be written or created, it is not required to allow this object to be set to readOnly, permanent, or nonVolatile."

OBJECT ipNetToPhysicalPhysAddress

MIN-ACCESS read-only

DESCRIPTION

"An agent is not required to provide write or create access to this object."

OBJECT ipNetToPhysicalType

MIN-ACCESS read-only

DESCRIPTION

"An agent is not required to provide write or create access to this object."

OBJECT ipv6RouterAdvertSpinLock

MIN-ACCESS read-only

DESCRIPTION

"An agent is not required to provide write access to this object. However, if an agent provides write access to any of the other objects in the ipv6RouterAdvertGroup, it SHOULD provide write access to this object as well."

OBJECT ipv6RouterAdvertSendAdverts

MIN-ACCESS read-only

DESCRIPTION

"An agent is not required to provide write access to this object."

OBJECT ipv6RouterAdvertMaxInterval

MIN-ACCESS read-only

DESCRIPTION

"An agent is not required to provide write access to this object."

OBJECT ipv6RouterAdvertMinInterval

MIN-ACCESS read-only

DESCRIPTION

"An agent is not required to provide write access to this object."

OBJECT ipv6RouterAdvertManagedFlag

MIN-ACCESS read-only

DESCRIPTION

"An agent is not required to provide write access to this object."

OBJECT ipv6RouterAdvertOtherConfigFlag
MIN-ACCESS read-only
DESCRIPTION
 "An agent is not required to provide write access to this
 object."

OBJECT ipv6RouterAdvertLinkMTU
MIN-ACCESS read-only
DESCRIPTION
 "An agent is not required to provide write access to this
 object."

OBJECT ipv6RouterAdvertReachableTime
MIN-ACCESS read-only
DESCRIPTION
 "An agent is not required to provide write access to this
 object."

OBJECT ipv6RouterAdvertRetransmitTime
MIN-ACCESS read-only
DESCRIPTION
 "An agent is not required to provide write access to this
 object."

OBJECT ipv6RouterAdvertCurHopLimit
MIN-ACCESS read-only
DESCRIPTION
 "An agent is not required to provide write access to this
 object."

OBJECT ipv6RouterAdvertDefaultLifetime
MIN-ACCESS read-only
DESCRIPTION
 "An agent is not required to provide write access to this
 object."

OBJECT ipv6RouterAdvertRowStatus
MIN-ACCESS read-only
DESCRIPTION
 "An agent is not required to provide write or create access
 to this object."

::= { ipMIBCompliances 2 }

-- units of conformance

ipv4GeneralGroup OBJECT-GROUP
 OBJECTS { ipForwarding, ipDefaultTTL, ipReasmTimeout }

```
STATUS      current
DESCRIPTION
    "The group of IPv4-specific objects for basic management of
    IPv4 entities."
 ::= { ipMIBGroups 3 }
```

ipv4IfGroup OBJECT-GROUP

```
OBJECTS { ipv4InterfaceReasmMaxSize, ipv4InterfaceEnableStatus,
           ipv4InterfaceRetransmitTime }
STATUS   current
DESCRIPTION
    "The group of IPv4-specific objects for basic management of
    IPv4 interfaces."
 ::= { ipMIBGroups 4 }
```

ipv6GeneralGroup2 OBJECT-GROUP

```
OBJECTS { ipv6IpForwarding, ipv6IpDefaultHopLimit }
STATUS   current
DESCRIPTION
    "The IPv6 group of objects providing for basic management of
    IPv6 entities."
 ::= { ipMIBGroups 5 }
```

ipv6IfGroup OBJECT-GROUP

```
OBJECTS { ipv6InterfaceReasmMaxSize,   ipv6InterfaceIdentifier,
           ipv6InterfaceEnableStatus,   ipv6InterfaceReachableTime,
           ipv6InterfaceRetransmitTime, ipv6InterfaceForwarding }
STATUS   current
DESCRIPTION
    "The group of IPv6-specific objects for basic management of
    IPv6 interfaces."
 ::= { ipMIBGroups 6 }
```

ipLastChangeGroup OBJECT-GROUP

```
OBJECTS { ipv4InterfaceTableLastChange,
           ipv6InterfaceTableLastChange,
           ipIfStatsTableLastChange }
STATUS   current
DESCRIPTION
    "The last change objects associated with this MIB.  These
    objects are optional for all agents.  They SHOULD be
    implemented on agents where it is possible to determine the
    proper values.  Where it is not possible to determine the
    proper values, for example when the tables are split amongst
    several sub-agents using AgentX, the agent MUST NOT
    implement these objects to return an incorrect or static
    value."
 ::= { ipMIBGroups 7 }
```

ipSystemStatsGroup OBJECT-GROUP

```
OBJECTS { ipSystemStatsInReceives,
           ipSystemStatsInOctets,
           ipSystemStatsInHdrErrors,
           ipSystemStatsInNoRoutes,
           ipSystemStatsInAddrErrors,
           ipSystemStatsInUnknownProtos,
           ipSystemStatsInTruncatedPkts,
           ipSystemStatsInForwDatagrams,
           ipSystemStatsReasmReqds,
           ipSystemStatsReasmOKs,
           ipSystemStatsReasmFails,
           ipSystemStatsInDiscards,
           ipSystemStatsInDelivers,
           ipSystemStatsOutRequests,
           ipSystemStatsOutNoRoutes,
           ipSystemStatsOutForwDatagrams,
           ipSystemStatsOutDiscards,
           ipSystemStatsOutFragReqds,
           ipSystemStatsOutFragOKs,
           ipSystemStatsOutFragFails,
           ipSystemStatsOutFragCreates,
           ipSystemStatsOutTransmits,
           ipSystemStatsOutOctets,
           ipSystemStatsInMcastPkts,
           ipSystemStatsInMcastOctets,
           ipSystemStatsOutMcastPkts,
           ipSystemStatsOutMcastOctets,
           ipSystemStatsDiscontinuityTime,
           ipSystemStatsRefreshRate }
```

```
STATUS current
```

```
DESCRIPTION
```

```
"IP system wide statistics."
```

```
::= { ipMIBGroups 8 }
```

ipv4SystemStatsGroup OBJECT-GROUP

```
OBJECTS { ipSystemStatsInBcastPkts, ipSystemStatsOutBcastPkts }
```

```
STATUS current
```

```
DESCRIPTION
```

```
"IPv4 only system wide statistics."
```

```
::= { ipMIBGroups 9 }
```

ipSystemStatsHCOctetGroup OBJECT-GROUP

```
OBJECTS { ipSystemStatsHCInOctets,
           ipSystemStatsHCOctets,
           ipSystemStatsHCInMcastOctets,
           ipSystemStatsHCOctets }
```

STATUS current

DESCRIPTION

"IP system wide statistics for systems that may overflow the standard octet counters within 1 hour."

::= { ipMIBGroups 10 }

ipSystemStatsHCPacketGroup OBJECT-GROUP

OBJECTS { ipSystemStatsHCInReceives,
ipSystemStatsHCInForwDatagrams,
ipSystemStatsHCInDelivers,
ipSystemStatsHCOutRequests,
ipSystemStatsHCOutForwDatagrams,
ipSystemStatsHCOutTransmits,
ipSystemStatsHCInMcastPkts,
ipSystemStatsHCOutMcastPkts

}

STATUS current

DESCRIPTION

"IP system wide statistics for systems that may overflow the standard packet counters within 1 hour."

::= { ipMIBGroups 11 }

ipv4SystemStatsHCPacketGroup OBJECT-GROUP

OBJECTS { ipSystemStatsHCInBcastPkts,
ipSystemStatsHCOutBcastPkts }

STATUS current

DESCRIPTION

"IPv4 only system wide statistics for systems that may overflow the standard packet counters within 1 hour."

::= { ipMIBGroups 12 }

ipIfStatsGroup OBJECT-GROUP

OBJECTS { ipIfStatsInReceives, ipIfStatsInOctets,
ipIfStatsInHdrErrors, ipIfStatsInNoRoutes,
ipIfStatsInAddrErrors, ipIfStatsInUnknownProtos,
ipIfStatsInTruncatedPkts, ipIfStatsInForwDatagrams,
ipIfStatsReasmReqds, ipIfStatsReasmOKs,
ipIfStatsReasmFails, ipIfStatsInDiscards,
ipIfStatsInDelivers, ipIfStatsOutRequests,
ipIfStatsOutForwDatagrams, ipIfStatsOutDiscards,
ipIfStatsOutFragReqds, ipIfStatsOutFragOKs,
ipIfStatsOutFragFails, ipIfStatsOutFragCreates,
ipIfStatsOutTransmits, ipIfStatsOutOctets,
ipIfStatsInMcastPkts, ipIfStatsInMcastOctets,
ipIfStatsOutMcastPkts, ipIfStatsOutMcastOctets,
ipIfStatsDiscontinuityTime, ipIfStatsRefreshRate }

STATUS current

DESCRIPTION

```
        "IP per-interface statistics."
 ::= { ipMIBGroups 13 }

ipv4IfStatsGroup OBJECT-GROUP
  OBJECTS { ipIfStatsInBcastPkts, ipIfStatsOutBcastPkts }
  STATUS current
  DESCRIPTION
    "IPv4 only per-interface statistics."
 ::= { ipMIBGroups 14 }

ipIfStatsHCOctetGroup OBJECT-GROUP
  OBJECTS { ipIfStatsHCInOctets, ipIfStatsHCOutOctets,
            ipIfStatsHCInMcastOctets, ipIfStatsHCOutMcastOctets }
  STATUS current
  DESCRIPTION
    "IP per-interfaces statistics for systems that include
     interfaces that may overflow the standard octet
     counters within 1 hour."
 ::= { ipMIBGroups 15 }

ipIfStatsHCPacketGroup OBJECT-GROUP
  OBJECTS { ipIfStatsHCInReceives, ipIfStatsHCInForwDatagrams,
            ipIfStatsHCInDelivers, ipIfStatsHCOutRequests,
            ipIfStatsHCOutForwDatagrams, ipIfStatsHCOutTransmits,
            ipIfStatsHCInMcastPkts, ipIfStatsHCOutMcastPkts }
  STATUS current
  DESCRIPTION
    "IP per-interfaces statistics for systems that include
     interfaces that may overflow the standard packet counters
     within 1 hour."
 ::= { ipMIBGroups 16 }

ipv4IfStatsHCPacketGroup OBJECT-GROUP
  OBJECTS { ipIfStatsHCInBcastPkts, ipIfStatsHCOutBcastPkts }
  STATUS current
  DESCRIPTION
    "IPv4 only per-interface statistics for systems that include
     interfaces that may overflow the standard packet counters
     within 1 hour."
 ::= { ipMIBGroups 17 }

ipAddressPrefixGroup OBJECT-GROUP
  OBJECTS { ipAddressPrefixOrigin,
            ipAddressPrefixOnLinkFlag,
            ipAddressPrefixAutonomousFlag,
            ipAddressPrefixAdvPreferredLifetime,
            ipAddressPrefixAdvValidLifetime }
  STATUS current
```

DESCRIPTION

"The group of objects for providing information about address prefixes used by this node."

::= { ipMIBGroups 18 }

ipAddressGroup OBJECT-GROUP

OBJECTS { ipAddressSpinLock, ipAddressIfIndex,
ipAddressType, ipAddressPrefix,
ipAddressOrigin, ipAddressStatus,
ipAddressCreated, ipAddressLastChanged,
ipAddressRowStatus, ipAddressStorageType }

STATUS current

DESCRIPTION

"The group of objects for providing information about the addresses relevant to this entity's interfaces."

::= { ipMIBGroups 19 }

ipNetToPhysicalGroup OBJECT-GROUP

OBJECTS { ipNetToPhysicalPhysAddress, ipNetToPhysicalLastUpdated,
ipNetToPhysicalType, ipNetToPhysicalState,
ipNetToPhysicalRowStatus }

STATUS current

DESCRIPTION

"The group of objects for providing information about the mappings of network address to physical address known to this node."

::= { ipMIBGroups 20 }

ipv6ScopeGroup OBJECT-GROUP

OBJECTS { ipv6ScopeZoneIndexLinkLocal,
ipv6ScopeZoneIndex3,
ipv6ScopeZoneIndexAdminLocal,
ipv6ScopeZoneIndexSiteLocal,
ipv6ScopeZoneIndex6,
ipv6ScopeZoneIndex7,
ipv6ScopeZoneIndexOrganizationLocal,
ipv6ScopeZoneIndex9,
ipv6ScopeZoneIndexA,
ipv6ScopeZoneIndexB,
ipv6ScopeZoneIndexC,
ipv6ScopeZoneIndexD }

STATUS current

DESCRIPTION

"The group of objects for managing IPv6 scope zones."

::= { ipMIBGroups 21 }

ipDefaultRouterGroup OBJECT-GROUP

OBJECTS { ipDefaultRouterLifetime, ipDefaultRouterPreference }

STATUS current

DESCRIPTION

"The group of objects for providing information about default routers known to this node."

::= { ipMIBGroups 22 }

ipv6RouterAdvertGroup OBJECT-GROUP

OBJECTS { ipv6RouterAdvertSpinLock,
ipv6RouterAdvertSendAdverts,
ipv6RouterAdvertMaxInterval,
ipv6RouterAdvertMinInterval,
ipv6RouterAdvertManagedFlag,
ipv6RouterAdvertOtherConfigFlag,
ipv6RouterAdvertLinkMTU,
ipv6RouterAdvertReachableTime,
ipv6RouterAdvertRetransmitTime,
ipv6RouterAdvertCurHopLimit,
ipv6RouterAdvertDefaultLifetime,
ipv6RouterAdvertRowStatus

}

STATUS current

DESCRIPTION

"The group of objects for controlling information advertised by IPv6 routers."

::= { ipMIBGroups 23 }

icmpStatsGroup OBJECT-GROUP

OBJECTS { icmpStatsInMsgs, icmpStatsInErrors,
icmpStatsOutMsgs, icmpStatsOutErrors,
icmpMsgStatsInPkts, icmpMsgStatsOutPkts }

STATUS current

DESCRIPTION

"The group of objects providing ICMP statistics."

::= { ipMIBGroups 24 }

--

-- Deprecated objects

--

ipInReceives OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS deprecated

DESCRIPTION

"The total number of input datagrams received from interfaces, including those received in error."

This object has been deprecated, as a new IP version-neutral

table has been added. It is loosely replaced by
ipSystemStatsInReceives."
 ::= { ip 3 }

ipInHdrErrors OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS deprecated
DESCRIPTION

"The number of input datagrams discarded due to errors in their IPv4 headers, including bad checksums, version number mismatch, other format errors, time-to-live exceeded, errors discovered in processing their IPv4 options, etc.

This object has been deprecated as a new IP version-neutral table has been added. It is loosely replaced by
ipSystemStatsInHdrErrors."

::= { ip 4 }

ipInAddrErrors OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS deprecated
DESCRIPTION

"The number of input datagrams discarded because the IPv4 address in their IPv4 header's destination field was not a valid address to be received at this entity. This count includes invalid addresses (e.g., 0.0.0.0) and addresses of unsupported Classes (e.g., Class E). For entities which are not IPv4 routers, and therefore do not forward datagrams, this counter includes datagrams discarded because the destination address was not a local address.

This object has been deprecated, as a new IP version-neutral table has been added. It is loosely replaced by
ipSystemStatsInAddrErrors."

::= { ip 5 }

ipForwDatagrams OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS deprecated
DESCRIPTION

"The number of input datagrams for which this entity was not their final IPv4 destination, as a result of which an attempt was made to find a route to forward them to that final destination. In entities which do not act as IPv4 routers, this counter will include only those packets which

were Source-Routed via this entity, and the Source-Route option processing was successful.

This object has been deprecated, as a new IP version-neutral table has been added. It is loosely replaced by ipSystemStatsInForwDatagrams."

::= { ip 6 }

ipInUnknownProtos OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS deprecated

DESCRIPTION

"The number of locally-addressed datagrams received successfully but discarded because of an unknown or unsupported protocol.

This object has been deprecated, as a new IP version-neutral table has been added. It is loosely replaced by ipSystemStatsInUnknownProtos."

::= { ip 7 }

ipInDiscards OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS deprecated

DESCRIPTION

"The number of input IPv4 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.

This object has been deprecated, as a new IP version-neutral table has been added. It is loosely replaced by ipSystemStatsInDiscards."

::= { ip 8 }

ipInDelivers OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS deprecated

DESCRIPTION

"The total number of input datagrams successfully delivered to IPv4 user-protocols (including ICMP).

This object has been deprecated as a new IP version neutral table has been added. It is loosely replaced by

```
        ipSystemStatsIndelivers."  
 ::= { ip 9 }
```

ipOutRequests OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS deprecated

DESCRIPTION

"The total number of IPv4 datagrams which local IPv4 user protocols (including ICMP) supplied to IPv4 in requests for transmission. Note that this counter does not include any datagrams counted in ipForwDatagrams.

This object has been deprecated, as a new IP version-neutral table has been added. It is loosely replaced by
ipSystemStatsOutRequests."

```
 ::= { ip 10 }
```

ipOutDiscards OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS deprecated

DESCRIPTION

"The number of output IPv4 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 ipForwDatagrams if any such packets met this (discretionary) discard criterion.

This object has been deprecated, as a new IP version-neutral table has been added. It is loosely replaced by
ipSystemStatsOutDiscards."

```
 ::= { ip 11 }
```

ipOutNoRoutes OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS deprecated

DESCRIPTION

"The number of IPv4 datagrams discarded because no route could be found to transmit them to their destination. Note that this counter includes any packets counted in ipForwDatagrams which meet this 'no-route' criterion. Note that this includes any datagrams which a host cannot route because all of its default routers are down.

This object has been deprecated, as a new IP version-neutral

table has been added. It is loosely replaced by
ipSystemStatsOutNoRoutes."
 ::= { ip 12 }

ipReasmReqds OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS deprecated
DESCRIPTION

"The number of IPv4 fragments received which needed to be
reassembled at this entity.

This object has been deprecated, as a new IP version-neutral
table has been added. It is loosely replaced by
ipSystemStatsReasmReqds."

::= { ip 14 }

ipReasmOKs OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS deprecated
DESCRIPTION

"The number of IPv4 datagrams successfully re-assembled.

This object has been deprecated, as a new IP version-neutral
table has been added. It is loosely replaced by
ipSystemStatsReasmOKs."

::= { ip 15 }

ipReasmFails OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS deprecated
DESCRIPTION

"The number of failures detected by the IPv4 re-assembly
algorithm (for whatever reason: timed out, errors, etc).
Note that this is not necessarily a count of discarded IPv4
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 object has been deprecated, as a new IP version-neutral
table has been added. It is loosely replaced by
ipSystemStatsReasmFails."

::= { ip 16 }

ipFragOKs OBJECT-TYPE
SYNTAX Counter32

MAX-ACCESS read-only
STATUS deprecated
DESCRIPTION

"The number of IPv4 datagrams that have been successfully fragmented at this entity.

This object has been deprecated, as a new IP version-neutral table has been added. It is loosely replaced by
ipSystemStatsOutFragOKs."

::= { ip 17 }

ipFragFails OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS deprecated
DESCRIPTION

"The number of IPv4 datagrams that have been discarded because they needed to be fragmented at this entity but could not be, e.g., because their Don't Fragment flag was set.

This object has been deprecated, as a new IP version-neutral table has been added. It is loosely replaced by
ipSystemStatsOutFragFails."

::= { ip 18 }

ipFragCreates OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS deprecated
DESCRIPTION

"The number of IPv4 datagram fragments that have been generated as a result of fragmentation at this entity.

This object has been deprecated as a new IP version neutral table has been added. It is loosely replaced by
ipSystemStatsOutFragCreates."

::= { ip 19 }

ipRoutingDiscards OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS deprecated
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.

This object was defined in pre-IPv6 versions of the IP MIB. It was implicitly IPv4 only, but the original specifications did not indicate this protocol restriction. In order to clarify the specifications, this object has been deprecated and a similar, but more thoroughly clarified, object has been added to the IP-FORWARD-MIB."

```
::= { ip 23 }
```

```
-- the deprecated IPv4 address table
```

```
ipAddrTable OBJECT-TYPE
```

```
SYNTAX      SEQUENCE OF IpAddrEntry
```

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

```
STATUS      deprecated
```

```
DESCRIPTION
```

```
"The table of addressing information relevant to this
entity's IPv4 addresses.
```

```
This table has been deprecated, as a new IP version-neutral
table has been added. It is loosely replaced by the
ipAddressTable although several objects that weren't deemed
useful weren't carried forward while another
(ipAdEntReasmMaxSize) was moved to the ipv4InterfaceTable."
```

```
::= { ip 20 }
```

```
ipAddrEntry OBJECT-TYPE
```

```
SYNTAX      IpAddrEntry
```

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

```
STATUS      deprecated
```

```
DESCRIPTION
```

```
"The addressing information for one of this entity's IPv4
addresses."
```

```
INDEX       { ipAdEntAddr }
```

```
::= { ipAddrTable 1 }
```

```
IpAddrEntry ::= SEQUENCE {
```

```
    ipAdEntAddr      IpAddress,
```

```
    ipAdEntIfIndex   INTEGER,
```

```
    ipAdEntNetMask    IpAddress,
```

```
    ipAdEntBcastAddr  INTEGER,
```

```
    ipAdEntReasmMaxSize INTEGER
```

```
}
```

```
ipAdEntAddr OBJECT-TYPE
```

```
SYNTAX      IpAddress
```

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

```
STATUS      deprecated
```

```
DESCRIPTION
```

"The IPv4 address to which this entry's addressing information pertains."
 ::= { ipAddrEntry 1 }

ipAdEntIfIndex OBJECT-TYPE

SYNTAX INTEGER (1..2147483647)

MAX-ACCESS read-only

STATUS deprecated

DESCRIPTION

"The index value which uniquely identifies the interface to which this entry is applicable. The interface identified by a particular value of this index is the same interface as identified by the same value of the IF-MIB's ifIndex."

::= { ipAddrEntry 2 }

ipAdEntNetMask OBJECT-TYPE

SYNTAX IpAddress

MAX-ACCESS read-only

STATUS deprecated

DESCRIPTION

"The subnet mask associated with the IPv4 address of this entry. The value of the mask is an IPv4 address with all the network bits set to 1 and all the hosts bits set to 0."

::= { ipAddrEntry 3 }

ipAdEntBcastAddr OBJECT-TYPE

SYNTAX INTEGER (0..1)

MAX-ACCESS read-only

STATUS deprecated

DESCRIPTION

"The value of the least-significant bit in the IPv4 broadcast address used for sending datagrams on the (logical) interface associated with the IPv4 address of this entry. For example, when the Internet standard all-ones broadcast address is used, the value will be 1. This value applies to both the subnet and network broadcast addresses used by the entity on this (logical) interface."

::= { ipAddrEntry 4 }

ipAdEntReasmMaxSize OBJECT-TYPE

SYNTAX INTEGER (0..65535)

MAX-ACCESS read-only

STATUS deprecated

DESCRIPTION

"The size of the largest IPv4 datagram which this entity can re-assemble from incoming IPv4 fragmented datagrams received on this interface."

::= { ipAddrEntry 5 }

-- the deprecated IPv4 Address Translation table

-- The Address Translation tables contain the IpAddress to
 -- "physical" address equivalences. Some interfaces do not
 -- use translation tables for determining address
 -- equivalences (e.g., DDN-X.25 has an algorithmic method);
 -- if all interfaces are of this type, then the Address
 -- Translation table is empty, i.e., has zero entries.

ipNetToMediaTable OBJECT-TYPE

SYNTAX SEQUENCE OF IpNetToMediaEntry

MAX-ACCESS not-accessible

STATUS deprecated

DESCRIPTION

"The IPv4 Address Translation table used for mapping from
 IPv4 addresses to physical addresses.

This table has been deprecated, as a new IP version-neutral
 table has been added. It is loosely replaced by the
 ipNetToPhysicalTable."

::= { ip 22 }

ipNetToMediaEntry OBJECT-TYPE

SYNTAX IpNetToMediaEntry

MAX-ACCESS not-accessible

STATUS deprecated

DESCRIPTION

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

INDEX { ipNetToMediaIfIndex,
 ipNetToMediaNetAddress }

::= { ipNetToMediaTable 1 }

IpNetToMediaEntry ::= SEQUENCE {

ipNetToMediaIfIndex INTEGER,
 ipNetToMediaPhysAddress PhysAddress,
 ipNetToMediaNetAddress IpAddress,
 ipNetToMediaType INTEGER

}

ipNetToMediaIfIndex OBJECT-TYPE

SYNTAX INTEGER (1..2147483647)

MAX-ACCESS read-create

STATUS deprecated

DESCRIPTION

"The interface on which this entry's equivalence is
 effective. The interface identified by a particular value
 of this index is the same interface as identified by the

same value of the IF-MIB's ifIndex.

This object predates the rule limiting index objects to a max access value of 'not-accessible' and so continues to use a value of 'read-create'."

```
::= { ipNetToMediaEntry 1 }
```

ipNetToMediaPhysAddress OBJECT-TYPE

SYNTAX PhysAddress (SIZE(0..65535))

MAX-ACCESS read-create

STATUS deprecated

DESCRIPTION

"The media-dependent 'physical' address. This object should return 0 when this entry is in the 'incomplete' state.

As the entries in this table are typically not persistent when this object is written the entity should not save the change to non-volatile storage. Note: a stronger requirement is not used because this object was previously defined."

```
::= { ipNetToMediaEntry 2 }
```

ipNetToMediaNetAddress OBJECT-TYPE

SYNTAX IpAddress

MAX-ACCESS read-create

STATUS deprecated

DESCRIPTION

"The IpAddress corresponding to the media-dependent 'physical' address.

This object predates the rule limiting index objects to a max access value of 'not-accessible' and so continues to use a value of 'read-create'."

```
::= { ipNetToMediaEntry 3 }
```

ipNetToMediaType OBJECT-TYPE

```
SYNTAX      INTEGER {
    other(1),          -- none of the following
    invalid(2),       -- an invalidated mapping
    dynamic(3),
    static(4)
}
```

MAX-ACCESS read-create

STATUS deprecated

DESCRIPTION

"The type of mapping.

Setting this object to the value invalid(2) has the effect

of invalidating the corresponding entry in the ipNetToMediaTable. That is, it effectively dis-associates 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 ipNetToMediaType object.

As the entries in this table are typically not persistent when this object is written the entity should not save the change to non-volatile storage. Note: a stronger requirement is not used because this object was previously defined."

::= { ipNetToMediaEntry 4 }

-- the deprecated ICMP group

icmpInMsgs OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS deprecated

DESCRIPTION

"The total number of ICMP messages which the entity received. Note that this counter includes all those counted by icmpInErrors.

This object has been deprecated, as a new IP version-neutral table has been added. It is loosely replaced by icmpStatsInMsgs."

::= { icmp 1 }

icmpInErrors OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS deprecated

DESCRIPTION

"The number of ICMP messages which the entity received but determined as having ICMP-specific errors (bad ICMP checksums, bad length, etc.).

This object has been deprecated, as a new IP version-neutral table has been added. It is loosely replaced by icmpStatsInErrors."

::= { icmp 2 }

`icmpInDestUnreachs OBJECT-TYPE``SYNTAX Counter32``MAX-ACCESS read-only``STATUS deprecated``DESCRIPTION`

"The number of ICMP Destination Unreachable messages received.

This object has been deprecated, as a new IP version-neutral table has been added. It is loosely replaced by a column in the icmpMsgStatsTable."

`::= { icmp 3 }``icmpInTimeExcds OBJECT-TYPE``SYNTAX Counter32``MAX-ACCESS read-only``STATUS deprecated``DESCRIPTION`

"The number of ICMP Time Exceeded messages received.

This object has been deprecated, as a new IP version-neutral table has been added. It is loosely replaced by a column in the icmpMsgStatsTable."

`::= { icmp 4 }``icmpInParmProbs OBJECT-TYPE``SYNTAX Counter32``MAX-ACCESS read-only``STATUS deprecated``DESCRIPTION`

"The number of ICMP Parameter Problem messages received.

This object has been deprecated, as a new IP version-neutral table has been added. It is loosely replaced by a column in the icmpMsgStatsTable."

`::= { icmp 5 }``icmpInSrcQuenchs OBJECT-TYPE``SYNTAX Counter32``MAX-ACCESS read-only``STATUS deprecated``DESCRIPTION`

"The number of ICMP Source Quench messages received.

This object has been deprecated, as a new IP version-neutral table has been added. It is loosely replaced by a column in the icmpMsgStatsTable."

`::= { icmp 6 }`

`icmpInRedirects OBJECT-TYPE``SYNTAX Counter32``MAX-ACCESS read-only``STATUS deprecated``DESCRIPTION``"The number of ICMP Redirect messages received.``This object has been deprecated, as a new IP version-neutral
 table has been added. It is loosely replaced by a column in
 the icmpMsgStatsTable."``::= { icmp 7 }``icmpInEchos OBJECT-TYPE``SYNTAX Counter32``MAX-ACCESS read-only``STATUS deprecated``DESCRIPTION``"The number of ICMP Echo (request) messages received.``This object has been deprecated, as a new IP version-neutral
 table has been added. It is loosely replaced by a column in
 the icmpMsgStatsTable."``::= { icmp 8 }``icmpInEchoReps OBJECT-TYPE``SYNTAX Counter32``MAX-ACCESS read-only``STATUS deprecated``DESCRIPTION``"The number of ICMP Echo Reply messages received.``This object has been deprecated, as a new IP version-neutral
 table has been added. It is loosely replaced by a column in
 the icmpMsgStatsTable."``::= { icmp 9 }``icmpInTimestamps OBJECT-TYPE``SYNTAX Counter32``MAX-ACCESS read-only``STATUS deprecated``DESCRIPTION``"The number of ICMP Timestamp (request) messages received.``This object has been deprecated, as a new IP version-neutral
 table has been added. It is loosely replaced by a column in
 the icmpMsgStatsTable."``::= { icmp 10 }`

`icmpInTimestampReps OBJECT-TYPE``SYNTAX Counter32``MAX-ACCESS read-only``STATUS deprecated``DESCRIPTION`

"The number of ICMP Timestamp Reply messages received.

This object has been deprecated, as a new IP version-neutral table has been added. It is loosely replaced by a column in the icmpMsgStatsTable."

`::= { icmp 11 }``icmpInAddrMasks OBJECT-TYPE``SYNTAX Counter32``MAX-ACCESS read-only``STATUS deprecated``DESCRIPTION`

"The number of ICMP Address Mask Request messages received.

This object has been deprecated, as a new IP version-neutral table has been added. It is loosely replaced by a column in the icmpMsgStatsTable."

`::= { icmp 12 }``icmpInAddrMaskReps OBJECT-TYPE``SYNTAX Counter32``MAX-ACCESS read-only``STATUS deprecated``DESCRIPTION`

"The number of ICMP Address Mask Reply messages received.

This object has been deprecated, as a new IP version-neutral table has been added. It is loosely replaced by a column in the icmpMsgStatsTable."

`::= { icmp 13 }``icmpOutMsgs OBJECT-TYPE``SYNTAX Counter32``MAX-ACCESS read-only``STATUS deprecated``DESCRIPTION`

"The total number of ICMP messages which this entity attempted to send. Note that this counter includes all those counted by icmpOutErrors.

This object has been deprecated, as a new IP version-neutral table has been added. It is loosely replaced by icmpStatsOutMsgs."

```
::= { icmp 14 }
```

```
icmpOutErrors OBJECT-TYPE
```

```
SYNTAX Counter32
```

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

```
STATUS deprecated
```

```
DESCRIPTION
```

"The number of ICMP messages which this entity did not send due to problems discovered within ICMP, such as a lack of buffers. This value should not include errors discovered outside the ICMP layer, such as the inability of IP to route the resultant datagram. In some implementations, there may be no types of error which contribute to this counter's value.

This object has been deprecated, as a new IP version-neutral table has been added. It is loosely replaced by icmpStatsOutErrors."

```
::= { icmp 15 }
```

```
icmpOutDestUnreachs OBJECT-TYPE
```

```
SYNTAX Counter32
```

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

```
STATUS deprecated
```

```
DESCRIPTION
```

"The number of ICMP Destination Unreachable messages sent.

This object has been deprecated, as a new IP version-neutral table has been added. It is loosely replaced by a column in the icmpMsgStatsTable."

```
::= { icmp 16 }
```

```
icmpOutTimeExcds OBJECT-TYPE
```

```
SYNTAX Counter32
```

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

```
STATUS deprecated
```

```
DESCRIPTION
```

"The number of ICMP Time Exceeded messages sent.

This object has been deprecated, as a new IP version-neutral table has been added. It is loosely replaced by a column in the icmpMsgStatsTable."

```
::= { icmp 17 }
```

```
icmpOutParmProbs OBJECT-TYPE
```

```
SYNTAX Counter32
```

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

```
STATUS deprecated
```

DESCRIPTION

"The number of ICMP Parameter Problem messages sent.

This object has been deprecated, as a new IP version-neutral table has been added. It is loosely replaced by a column in the icmpMsgStatsTable."

::= { icmp 18 }

icmpOutSrcQuenchs OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS deprecated

DESCRIPTION

"The number of ICMP Source Quench messages sent.

This object has been deprecated, as a new IP version-neutral table has been added. It is loosely replaced by a column in the icmpMsgStatsTable."

::= { icmp 19 }

icmpOutRedirects OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS deprecated

DESCRIPTION

"The number of ICMP Redirect messages sent. For a host, this object will always be zero, since hosts do not send redirects.

This object has been deprecated, as a new IP version-neutral table has been added. It is loosely replaced by a column in the icmpMsgStatsTable."

::= { icmp 20 }

icmpOutEchos OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS deprecated

DESCRIPTION

"The number of ICMP Echo (request) messages sent.

This object has been deprecated, as a new IP version-neutral table has been added. It is loosely replaced by a column in the icmpMsgStatsTable."

::= { icmp 21 }

icmpOutEchoReps OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only
STATUS deprecated
DESCRIPTION

"The number of ICMP Echo Reply messages sent.

This object has been deprecated, as a new IP version-neutral table has been added. It is loosely replaced by a column in the icmpMsgStatsTable."

::= { icmp 22 }

icmpOutTimestamps OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS deprecated
DESCRIPTION

"The number of ICMP Timestamp (request) messages sent.

This object has been deprecated, as a new IP version-neutral table has been added. It is loosely replaced by a column in the icmpMsgStatsTable."

::= { icmp 23 }

icmpOutTimestampReps OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS deprecated
DESCRIPTION

"The number of ICMP Timestamp Reply messages sent.

This object has been deprecated, as a new IP version-neutral table has been added. It is loosely replaced by a column in the icmpMsgStatsTable."

::= { icmp 24 }

icmpOutAddrMasks OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS deprecated
DESCRIPTION

"The number of ICMP Address Mask Request messages sent.

This object has been deprecated, as a new IP version-neutral table has been added. It is loosely replaced by a column in the icmpMsgStatsTable."

::= { icmp 25 }

icmpOutAddrMaskReps OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only
 STATUS deprecated
 DESCRIPTION

"The number of ICMP Address Mask Reply messages sent.

This object has been deprecated, as a new IP version-neutral table has been added. It is loosely replaced by a column in the icmpMsgStatsTable."

::= { icmp 26 }

-- deprecated conformance information
 -- deprecated compliance statements

ipMIBCompliance MODULE-COMPLIANCE
 STATUS deprecated
 DESCRIPTION

"The compliance statement for systems that implement only IPv4. For version-independence, this compliance statement is deprecated in favor of ipMIBCompliance2."

MODULE -- this module
 MANDATORY-GROUPS { ipGroup,
 icmpGroup }
 ::= { ipMIBCompliances 1 }

-- deprecated units of conformance

ipGroup OBJECT-GROUP

OBJECTS {	ipForwarding,	ipDefaultTTL,
	ipInReceives,	ipInHdrErrors,
	ipInAddrErrors,	ipForwDatagrams,
	ipInUnknownProtos,	ipInDiscards,
	ipInDelivers,	ipOutRequests,
	ipOutDiscards,	ipOutNoRoutes,
	ipReasmTimeout,	ipReasmReqds,
	ipReasmOKs,	ipReasmFails,
	ipFragOKs,	ipFragFails,
	ipFragCreates,	ipAdEntAddr,
	ipAdEntIfIndex,	ipAdEntNetMask,
	ipAdEntBcastAddr,	ipAdEntReasmMaxSize,
	ipNetToMediaIfIndex,	ipNetToMediaPhysAddress,
	ipNetToMediaNetAddress,	ipNetToMediaType,
	ipRoutingDiscards	

}

STATUS deprecated
 DESCRIPTION

"The ip group of objects providing for basic management of IP entities, exclusive of the management of IP routes.

As part of the version independence, this group has been deprecated. "

```
 ::= { ipMIBGroups 1 }
```

icmpGroup OBJECT-GROUP

```
OBJECTS { icmpInMsgs,          icmpInErrors,
           icmpInDestUnreachs, icmpInTimeExcds,
           icmpInParmProbs,    icmpInSrcQuenchs,
           icmpInRedirects,    icmpInEchos,
           icmpInEchoReps,     icmpInTimestamps,
           icmpInTimestampReps, icmpInAddrMasks,
           icmpInAddrMaskReps, icmpOutMsgs,
           icmpOutErrors,      icmpOutDestUnreachs,
           icmpOutTimeExcds,   icmpOutParmProbs,
           icmpOutSrcQuenchs,  icmpOutRedirects,
           icmpOutEchos,       icmpOutEchoReps,
           icmpOutTimestamps,  icmpOutTimestampReps,
           icmpOutAddrMasks,   icmpOutAddrMaskReps }
```

STATUS deprecated

DESCRIPTION

"The icmp group of objects providing ICMP statistics.

As part of the version independence, this group has been deprecated. "

```
 ::= { ipMIBGroups 2 }
```

END

6. Previous Work

This document contains objects modified from RFC 1213 [11], RFC 2011 [12], RFC 2465 [13], and RFC 2466 [14].

7. References

7.1. Normative References

- [1] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.
- [2] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.
- [3] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

- [4] Narten, T., Nordmark, E., and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, December 1998.
- [5] Thomson, S. and T. Narten, "IPv6 Stateless Address Autoconfiguration", RFC 2462, December 1998.
- [6] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.
- [7] Daniele, M., Haberman, B., Routhier, S., and J. Schoenwaelder, "Textual Conventions for Internet Network Addresses", RFC 4001, February 2005.
- [8] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, November 2005.

7.2. Informative References

- [9] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.
- [10] Plummer, D., "Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware", STD 37, RFC 826, November 1982.
- [11] McCloghrie, K. and M. Rose, "Management Information Base for Network Management of TCP/IP-based internets:MIB-II", STD 17, RFC 1213, March 1991.
- [12] McCloghrie, K., "SNMPv2 Management Information Base for the Internet Protocol using SMIV2", RFC 2011, November 1996.
- [13] Haskin, D. and S. Onishi, "Management Information Base for IP Version 6: Textual Conventions and General Group", RFC 2465, December 1998.
- [14] Haskin, D. and S. Onishi, "Management Information Base for IP Version 6: ICMPv6 Group", RFC 2466, December 1998.
- [15] Narten, T. and R. Draves, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 3041, January 2001.
- [16] Haberman, B., "IP Forwarding Table MIB", RFC 4292, April 2006.

[17] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

8. Security Considerations

There are a number of management objects defined in this MIB module with a MAX-ACCESS clause of read-write and/or read-create. Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations. These are the tables and objects and their sensitivity/vulnerability:

ipForwarding and ipv6IpForwarding - these objects allow a manager to enable or disable the routing functions on the entity. By disabling the routing functions, an attacker would possibly be able to deny service to users. By enabling the routing functions, an attacker could open a conduit into an area. This might result in the area providing transit for packets it shouldn't or might allow the attacker access to the area bypassing security safeguards.

ipDefaultTTL and ipv6IpDefaultHopLimit - these objects allow a manager to determine the diameter of the valid area for a packet. By decreasing the value of these objects, an attacker could cause packets to be discarded before reaching their destinations.

ipv4InterfaceEnableStatus and ipv6InterfaceEnableStatus - these objects allow a manager to enable or disable IPv4 and IPv6 on a specific interface. By enabling a protocol on an interface, an attacker might be able to create an unsecured path into a node (or through it if routing is also enabled). By disabling a protocol on an interface, an attacker might be able to force packets to be routed through some other interface or deny access to some or all of the network via that protocol.

ipAddressTable - the objects in this table specify the addresses in use on this node. By modifying this information, an attacker can cause a node to either ignore messages destined to it or accept (at least at the IP layer) messages it would otherwise ignore. The use of filtering or security associations may reduce the potential damage in the latter case.

ipv6RouterAdvertTable - the objects in this table specify the information that a router should propagate in its routing advertisement messages. By modifying this information, an attacker can interfere with the auto-configuration of all hosts on the link. Most modifications to this table will result in a

denial of service to some or all hosts on the link. However two objects, `ipv6RouterAdvertManagedFlag` and `ipv6RouterAdvertOtherConfigFlag`, indicate if a host should acquire configuration information from some other source. By enabling these, an attacker might be able to cause a host to retrieve its configuration information from a compromised source.

`ipNetToPhysicalPhysAddress` and `ipNetToPhysicalType` - these objects specify information used to translate a network (IP) address into a media dependent address. By modifying these objects, an attacker could disable communication with a node or divert messages from one node to another. However, the attacker may be able to carry out a similar attack by simply responding to the ARP or ND request made by the target node.

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

These are the tables and objects and their sensitivity/vulnerability:

Essentially, all of the objects in this MIB could be considered sensitive as they report on the status of the IP modules within a system. However, the `ipSystemStatsTable`, `ipIfStatsTable`, and `ipAddressTable` are likely to be of most interest to an attacker. The statistics tables supply information about the quantity and type of traffic this node is processing and, especially for transit providers, may be considered sensitive. The address table provides a convenient list of all addresses in use by this node. Each address in isolation is unremarkable, however, the total list would allow an attacker to correlate otherwise unrelated traffic. For example, an attacker might be able to correlate an RFC 3041 [15] private address with known public addresses, thus circumventing the intentions of RFC 3041.

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

It is RECOMMENDED that implementers consider the security features as provided by the SNMPv3 framework (see [9], section 8), including full support for the SNMPv3 cryptographic mechanisms (for authentication and privacy).

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

9. Acknowledgements

Reviews and other contributions were made by:

Dario Acornero, Cisco Systems
Mike MacFaden, VMWare
Keith McCloghrie, Cisco Systems
Juergen Schoenwalder, TU Braunschweig
Margaret Wasserman, Devicescape

10. Authors

This document was written by the IPv6 MIB revision design team:

Bill Fenner, AT&T Labs -- Research
EMail: fenner@research.att.com

Brian Haberman
EMail: brian@innovationslab.net

Shawn A. Routhier
EMail: sar@iwl.com

Dave Thaler, Microsoft
EMail: dthaler@microsoft.com

This document updates parts of the MIBs from several other documents. RFC 2011 is the previous update to the IP MIB. RFC 2465 and RFC 2466 are the first versions that specified IPv6 addresses and information.

RFC 2011:
Keith McCloghrie, Cisco Systems (Editor)

RFC 2465 and RFC 2466:
Dimitry Haskin, Bay Networks

Steve Onishi, Bay Networks

Editor's Contact Information

Shawn A. Routhier
Interworking Labs
108 Whispering Pines Dr. Suite 235
Scotts Valley, CA 95066
USA

EMail: sar@iwl.com

Full Copyright Statement

Copyright (C) The Internet Society (2006).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM 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.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).

