

Network Working Group  
Request for Comments: 4008  
Category: Standards Track

R. Rohit  
Mascon Global Limited  
P. Srisuresh  
Caymas Systems, Inc.  
R. Raghunarayan  
N. Pai  
Cisco Systems, Inc.  
C. Wang  
Bank One Corp  
March 2005

## Definitions of Managed Objects for Network Address Translators (NAT)

### 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 (2005).

### Abstract

This memo defines a portion of the Management Information Base (MIB) for devices implementing Network Address Translator (NAT) function. This MIB module may be used for configuration as well as monitoring of a device capable of NAT function.

## Table of Contents

|                                                               |    |
|---------------------------------------------------------------|----|
| 1. Introduction .....                                         | 2  |
| 2. The Internet-Standard Management Framework .....           | 2  |
| 3. Terminology .....                                          | 3  |
| 4. Overview .....                                             | 4  |
| 4.1. natInterfaceTable.....                                   | 4  |
| 4.2. natAddrMapTable.....                                     | 5  |
| 4.3. Default Timeouts, Protocol Table, and Other Scalars..... | 6  |
| 4.4. natAddrBindTable and natAddrPortBindTable.....           | 6  |
| 4.5. natSessionTable.....                                     | 6  |
| 4.6. RFC 3489 NAT Variations, NAT Session and Bind Tables...  | 7  |
| 4.7. Notifications.....                                       | 7  |
| 4.8. Relation Among Tables.....                               | 8  |
| 4.9. Configuration via the MIB.....                           | 8  |
| 4.10. Relationship to Interface MIB.....                      | 9  |
| 5. Definitions .....                                          | 9  |
| 6. Acknowledgements .....                                     | 59 |
| 7. Security Considerations .....                              | 59 |
| 8. References .....                                           | 60 |
| Authors' Addresses .....                                      | 62 |
| Full Copyright Statement.....                                 | 64 |

## 1. Introduction

This memo defines a portion of the Management Information Base (MIB) for devices implementing NAT function. This MIB module may be used for configuration and monitoring of a device capable of NAT function. NAT types and their characteristics are defined in [RFC2663]. Traditional NAT function, in particular is defined in [RFC3022]. This MIB does not address the firewall functions and must not be used for configuring or monitoring these. Section 2 provides references to the SNMP management framework, which was used as the basis for the MIB module definition. Section 3 describes the terms used throughout the document. Section 4 provides an overview of the key objects, their inter-relationship, and how the MIB module may be used to configure and monitor a NAT device. Lastly, section 5 has the complete NAT MIB definition.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

## 2. 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 [RFC3410].

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 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

### 3. Terminology

Definitions for a majority of the terms used throughout the document may be found in RFC 2663 [RFC2663]. Additional terms that further classify NAPT implementations are defined in RFC 3489 [RFC3489]. Listed below are terms used in this document.

**Address realm** - An address realm is a realm of unique network addresses that are routable within the realm. For example, an enterprise address realm could be constituted of private IP addresses in the ranges specified in RFC 1918 [RFC1918], which are routable within the enterprise, but not across the Internet. A public realm is constituted of globally unique network addresses.

**Symmetric NAT** - Symmetric NAT, as defined in RFC 3489 [RFC3489], is a variation of Network Address Port Translator (NAPT). Symmetric NAT does not use port bind for translation across all sessions originating from the same private host. Instead, it assigns a new public port to each new session, irrespective of whether the new session used the same private end-point as before.

**Bind or Binding** - Several variations of the term 'Bind' (or 'Binding') are used throughout the document. Address Bind (or Address Binding) is a tuple of (Private IP address, Public IP Address) used for translating an IP address end-point in IP packets. Port Bind (or, Port Binding, or Address Port Bind, or Address Port Binding) is a tuple of (transport protocol, Private IP address, Private port, Public IP Address, Public port) used for translating a port end-point tuple of (transport protocol, IP address, port). Bind is used to refer to either Address Bind or Port Bind. Bind Mode identifies whether a bind is Address Bind or Port Bind.

**NAT Session** - A NAT session is an association between a session as seen in the private realm and a session as seen in the public realm, by virtue of NAT translation. If a session in the private realm were to be represented as (PrivateSrcAddr, PrivateDstAddr, TransportProtocol, PrivateSrcPort, PrivateDstPort) and the same session in the public realm were to be represented as (PublicSrcAddr,

PublicDstAddr, TransportProtocol, PublicSrcPort, PublicDstPort), the NAT session will provide the translation glue between the two session representations. NAT sessions in the document are restricted to sessions based on TCP and UDP only. In the future, NAT sessions may be extended to be based on other transport protocols such as SCTP, UDP-lite and DCCP.

The terms 'local' and 'private' are used interchangeably throughout the document when referring to private networks, IP addresses, and ports. Likewise, the terms 'global' and 'public' are used interchangeably when referring to public networks, IP addresses, and ports.

#### 4. Overview

NAT MIB is configurable on a per-interface basis and depends in several parts on the IF-MIB [RFC2863].

NAT MIB requires that an interface for which NAT is configured be connected to either a private or a public realm. The realm association of the interface plays an important role in the definition of address maps for the interface. An address map entry identifies the orientation of the session (inbound or outbound to the interface) for which the entry may be used for NAT translation. The address map entry also identifies the end-point of the session that must be subject to translation. An SNMP Textual-Convention 'NatTranslationEntity' is defined to capture this important characteristic that combines session orientation and applicable session endpoint for translation.

An address map may consist of static or dynamic entries. NAT creates static binds from a static address map entry. Each static bind has a direct one-to-one relationship with a static address map entry. NAT creates dynamic binds from a dynamic address map entry upon seeing the first packet of a new session.

The following subsections define the key objects used in NAT MIB, their inter-relationship, and how to configure a NAT device using the MIB module.

##### 4.1. natInterfaceTable

natInterfaceTable is defined in the MIB module to configure interface specific realm type and the NAT services enabled for the interface. natInterfaceTable is indexed by ifIndex and also includes interface specific NAT statistics.

The first step for an operator in configuring a NAT device is determining the interface over which NAT service is to be configured. When NAT service is operational, translated packets traverse the NAT device by ingressing on a private interface and egressing on a public interface or vice versa. An operator may configure the NAT service on either the public interface or the private interface in the traversal path.

As the next step, the operator must identify the NAT service(s) desired for the interface. The operator may configure one or more NAT services on the same interface. The MIB module identifies four types of NAT services: Basic NAT, NAPT, twice NAT and bidirectional NAT. These are NAT varieties as defined in RFC 2663 [RFC2663]. Note that RFC 3489 [RFC3489] further classifies NAPT implementations based on the behavior exhibited by the NAPT devices from different vendors. However, the MIB module does not explicitly distinguish between the NAPT implementations. NAPT implementations may be distinguished between one another by monitoring the BIND and NAT Session objects generated by the NAT device as described in section 4.6.

#### 4.2. natAddrMapTable

natAddrMapTable is defined in the MIB module to configure address maps on a per-interface basis. natAddrMapTable is indexed by the tuple of (ifIndex, natAddrMapIndex). The same table is also used to collect Statistics for the address map entries. Address maps are key to NAT configuration. An operator may configure one or more address map entries per interface. NAT looks up address map entries in the order in which they are defined to determine the translation function at the start of each new session traversing the interface. An address map may consist of static or dynamic entries. A static address map entry has a direct one-to-one relationship with binds. NAT will dynamically create binds from a dynamic address map entry.

The operator must be careful in selecting address map entries for an interface based on the interface realm-type and the type of NAT service desired. The operator can be amiss in the selection of address map entries when not paying attention to the associated interface characteristics defined in natInterfaceTable (described in section 4.1). For example, say the operator wishes to configure a NAPT map entry on an interface of a NAT device. If the operator chooses to configure the NAPT map entry on a public interface (i.e., interface realm-type is public), the operator should set the TranslationEntity of the NAPT address map entry to be outboundSrcEndPoint. On the other hand, if the operator chooses to configure the NAPT map entry on a private interface (i.e., interface realm-type is private), the operator should set the TranslationEntity of the NAPT address map entry to be InboundSrcEndPoint.

#### 4.3. Default Timeouts, Protocol Table, and Other Scalars

DefTimeouts is defined in the MIB module to configure idle Bind timeout and IP protocol specific idle NAT session timeouts. The timeouts defined are global to the system and are not interface specific.

Protocol specific statistics are maintained in natProtocolTable, which is indexed by the protocol type.

The scalars natAddrBindNumberOfEntries and natAddrPortBindNumberOfEntries hold the number of entries that currently exist in the Address Bind and the Address Port Bind tables, respectively.

The generation of natPacketDiscard notifications can be configured by using the natNotifThrottlingInterval scalar MIB object.

#### 4.4. natAddrBindTable and natAddrPortBindTable

Two Bind tables, natAddrBindTable and natAddrPortBindTable, are defined to hold the bind entries. Entries are derived from the address map table and are not configurable. natAddrBindTable contains Address Binds, and natAddrPortBindTable contains Address Port Binds. natAddrBindTable is indexed by the tuple of (ifIndex, LocalAddrType, LocalAddr). natAddrPortBindTable is indexed by the tuple of (ifIndex, LocalAddrType, LocalAddr, LocalPort, Protocol). These tables also maintain bind specific statistics. A Symmetric NAT will have no entries in the Bind tables.

#### 4.5. natSessionTable

natSessionTable is defined to hold NAT session entries. NAT session entries are derived from NAT Binds (except in the case of Symmetric NAT) and are not configurable.

The NAT session provides the necessary translation glue between two session representations of the same end-to-end session; that is, a session as seen in the private realm and in the public realm. Session orientation (inbound or outbound) is determined from the orientation of the first packet traversing the NAT interface. Address map entries and bind entries on the interface determine whether a session is subject to NAT translation. One or both endpoints of a session may be subject to translation.

With the exception of symmetric NAT, all other NAT functions use end-point specific bind to perform individual end-point translations. Multiple NAT sessions would use the same bind as long as they share

the same endpoint. Symmetric NAT does not retain a consistent port bind across multiple sessions using the same endpoint. For this reason, the bind identifier for a NAT session in symmetric NAT is set to zero. natSessionTable is indexed by the tuple of (ifIndex, natSessionIndex). Statistics for NAT sessions are also maintained in the same table.

#### 4.6. RFC 3489 NAPT Variations, NAT Session and Bind Tables

[RFC3489] defines four variations of NAPT - Full Cone, Restricted Cone, Port Restricted Cone, and Symmetric NAT. These can be differentiated in the NAT MIB based on different values for the objects in the session and the bind tables, as indicated below.

In a Port Restricted Cone NAT, NAT Session objects will contain a non-zero PrivateSrcEPBindId object. Further, all address and port objects within a NAT session will have non-zero values (i.e., no wildcard matches).

An Address Restricted Cone NAT may have been implemented in the same way as a Port Restricted Cone NAT, except that the UDP NAT Sessions may use ANY match on PrivateDstPort and PublicDstPort objects; i.e., PrivateDstPort and PublicDstPort objects within a NAT session may be set to zero.

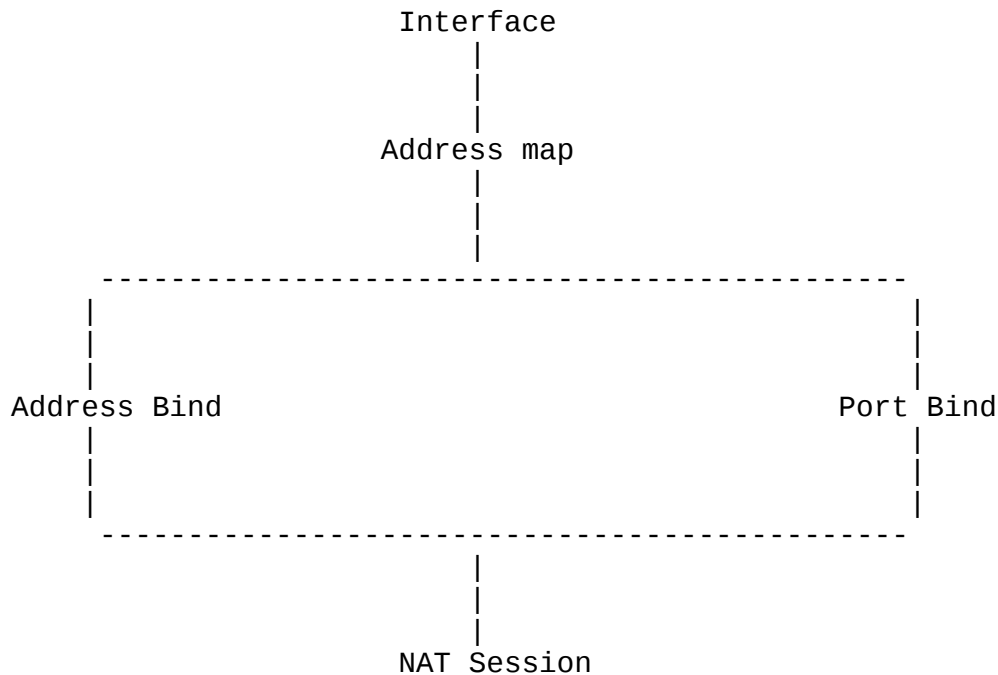
A Full Cone NAT may have also been implemented in the same way as a Port Restricted Cone NAT, except that the UDP NAT Sessions may use ANY match on PrivateDstAddr, PrivateDstPort, PublicDstAddr, and PublicDstPort objects. Within a NAT Session, all four of these objects may be set to zero. Alternately, all address and port objects within a NAT Session may have non-zero values, yet the TranslationEntity of the PrivateSrcEPBindId for the NAT Sessions may be set bi-directionally, i.e., as a bit mask of (outboundSrcEndPoint and inboundDstEndPoint) or (inboundSrcEndPoint and outboundDstEndPoint), depending on the interface realm type. Lastly, a Symmetric NAT does not maintain Port Bindings. As such, the NAT Session objects will have the PrivateSrcEPBindId set to zero.

#### 4.7. Notifications

natPacketDiscard notifies the end user/manager of packets being discarded due to lack of address mappings.

#### 4.8. Relation Among Tables

The association between the various NAT tables can be represented as follows:



All NAT functions, with the exception of Symmetric NAT, use Bind(s) to provide the glue necessary for a NAT Session. `natSessionPrivateSrcEPBindId` and `natSessionPrivateDstEPBindId` objects represent the endpoint Binds used by NAT Sessions.

#### 4.9. Configuration via the MIB

Sections 4.1 and 4.2 and part of section 4.3 refer to objects that are configurable on a NAT device. NAT derives Address Bind and Address Port Bind entries from the Address Map table. Hence, an Address Bind or an Address Port Bind entry must not exist without an associated entry in the Address Map table.

Further, NAT derives NAT session entries from NAT Binds, except in the case of symmetric NAT, which derives translation parameters for a NAT session directly from an address map entry. Hence, with the exception of Symmetric NAT, a NAT session entry must not exist in the NAT Session table without a corresponding bind.

A Management station may use the following steps to configure entries in the NAT-MIB:

- Create an entry in the natInterfaceTable specifying the value of ifIndex as the interface index of the interface on which NAT is being configured. Specify appropriate values, as applicable, for the other objects (e.g., natInterfaceRealm, natInterfaceServiceType) in the table (refer to Section 4.1).
- Create one or more address map entries sequentially in reduced order of priority in the natAddrMapTable, specifying the value of ifIndex to be the same for all entries. The ifIndex specified would be the same as that specified for natInterfaceTable (refer to Section 4.2).
- Configure the maximum permitted idle time duration for BINDs and TCP, UDP, and ICMP protocol sessions by setting the relevant scalars in natDefTimeouts object (refer to Section 4.3).

#### 4.10. Relationship to Interface MIB

The natInterfaceTable specifies the NAT configuration attributes on each interface. The concept of "interface" is as defined by InterfaceIndex/ifIndex of the IETF Interfaces MIB [RFC2863].

### 5. Definitions

This MIB module IMPORTs objects from RFCs 2578 [RFC2578], 2579 [RFC2579], 2580 [RFC2580], 2863 [RFC2863], 3411 [RFC3411], and 4001 [RFC4001]. It also refers to information in RFCs 792 [RFC792], 2463 [RFC2463], and 3413 [RFC3413].

NAT-MIB DEFINITIONS ::= BEGIN

#### IMPORTS

```
MODULE-IDENTITY,  
OBJECT-TYPE,  
Integer32,  
Unsigned32,  
Gauge32,  
Counter64,  
TimeTicks,  
mib-2,  
NOTIFICATION-TYPE  
    FROM SNMPv2-SMI  
TEXTUAL-CONVENTION,  
StorageType,  
RowStatus
```

```
        FROM SNMPv2-TC
MODULE-COMPLIANCE,
NOTIFICATION-GROUP,
OBJECT-GROUP
        FROM SNMPv2-CONF
ifIndex,
ifCounterDiscontinuityGroup
        FROM IF-MIB
SnmpAdminString
        FROM SNMP-FRAMEWORK-MIB
InetAddressType,
InetAddress,
InetPortNumber
        FROM INET-ADDRESS-MIB;
```

## natMIB MODULE-IDENTITY

```
LAST-UPDATED "200503210000Z"
ORGANIZATION "IETF Transport Area"
CONTACT-INFO
```

```
"
    Rohit
    Mascon Global Limited
    #59/2 100 ft Ring Road
    Banashankari II Stage
    Bangalore 560 070
    India
    Phone: +91 80 2679 6227
    Email: rrohit74@hotmail.com
```

```
    P. Srisuresh
    Caymas Systems, Inc.
    1179-A North McDowell Blvd.
    Petaluma, CA 94954
    Tel: (707) 283-5063
    Email: srisuresh@yahoo.com
```

```
    Rajiv Raghunarayan
    Cisco Systems Inc.
    170 West Tasman Drive
    San Jose, CA 95134
    Phone: +1 408 853 9612
    Email: raraghun@cisco.com
```

```
    Nalinaksh Pai
    Cisco Systems, Inc.
    Prestige Waterford
    No. 9, Brunton Road
    Bangalore - 560 025
```

India  
 Phone: +91 80 532 1300  
 Email: npai@cisco.com

Cliff Wang  
 Information Security  
 Bank One Corp  
 1111 Polaris Pkwy  
 Columbus, OH 43240  
 Phone: +1 614 213 6117  
 Email: cliffwang2000@yahoo.com

"

#### DESCRIPTION

"This MIB module defines the generic managed objects for NAT.

Copyright (C) The Internet Society (2005). This version of this MIB module is part of RFC 4008; see the RFC itself for full legal notices."

REVISION "200503210000Z" -- 21th March 2005

#### DESCRIPTION

"Initial version, published as RFC 4008."

::= { mib-2 123 }

natMIBObjects OBJECT IDENTIFIER ::= { natMIB 1 }

NatProtocolType ::= TEXTUAL-CONVENTION

STATUS current

#### DESCRIPTION

"A list of protocols that support the network address translation. Inclusion of the values is not intended to imply that those protocols need to be supported. Any change in this TEXTUAL-CONVENTION should also be reflected in the definition of NatProtocolMap, which is a BITS representation of this."

SYNTAX INTEGER {  
     none (1), -- not specified  
     other (2), -- none of the following  
     icmp (3),  
     udp (4),  
     tcp (5)  
 }

NatProtocolMap ::= TEXTUAL-CONVENTION

STATUS current

#### DESCRIPTION

"A bitmap of protocol identifiers that support

the network address translation. Any change in this TEXTUAL-CONVENTION should also be reflected in the definition of NatProtocolType."

```
SYNTAX BITS {
    other (0),
    icmp (1),
    udp (2),
    tcp (3)
}
```

```
NatAddrMapId ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS current
    DESCRIPTION
        "A unique id that is assigned to each address map
        by a NAT enabled device."
    SYNTAX Unsigned32 (1..4294967295)
```

```
NatBindIdOrZero ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS current
    DESCRIPTION
        "A unique id that is assigned to each bind by
        a NAT enabled device. The bind id will be zero
        in the case of a Symmetric NAT."
    SYNTAX Unsigned32 (0..4294967295)
```

```
NatBindId ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS current
    DESCRIPTION
        "A unique id that is assigned to each bind by
        a NAT enabled device."
    SYNTAX Unsigned32 (1..4294967295)
```

```
NatSessionId ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS current
    DESCRIPTION
        "A unique id that is assigned to each session by
        a NAT enabled device."
    SYNTAX Unsigned32 (1..4294967295)
```

```
NatBindMode ::= TEXTUAL-CONVENTION
    STATUS current
    DESCRIPTION
        "An indication of whether the bind is
        an address bind or an address port bind."
```

```

SYNTAX    INTEGER {
                addressBind (1),
                addressPortBind (2)
            }

```

NatAssociationType ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"An indication of whether the association is static or dynamic."

```

SYNTAX    INTEGER {
                static (1),
                dynamic (2)
            }

```

NatTranslationEntity ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"An indication of a) the direction of a session for which an address map entry, address bind or port bind is applicable, and b) the entity (source or destination) within the session that is subject to translation."

```

SYNTAX    BITS {
                inboundSrcEndPoint (0),
                outboundDstEndPoint(1),
                inboundDstEndPoint (2),
                outboundSrcEndPoint(3)
            }

```

--

-- Default Values for the Bind and NAT Protocol Timers

--

natDefTimeouts OBJECT IDENTIFIER ::= { natMIBObjects 1 }

natNotifCtrl OBJECT IDENTIFIER ::= { natMIBObjects 2 }

--

-- Address Bind and Port Bind related NAT configuration

--

natBindDefIdleTimeout OBJECT-TYPE

SYNTAX Unsigned32 (0..4294967295)

UNITS "seconds"

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The default Bind (Address Bind or Port Bind) idle timeout parameter.

If the agent is capable of storing non-volatile configuration, then the value of this object must be restored after a re-initialization of the management system."

DEFVAL { 0 }  
::= { natDefTimeouts 1 }

--  
-- UDP related NAT configuration  
--

natUdpDefIdleTimeout OBJECT-TYPE

SYNTAX Unsigned32 (1..4294967295)  
UNITS "seconds"  
MAX-ACCESS read-write  
STATUS current  
DESCRIPTION

"The default UDP idle timeout parameter.

If the agent is capable of storing non-volatile configuration, then the value of this object must be restored after a re-initialization of the management system."

DEFVAL { 300 }  
::= { natDefTimeouts 2 }

--  
-- ICMP related NAT configuration  
--

natIcmpDefIdleTimeout OBJECT-TYPE

SYNTAX Unsigned32 (1..4294967295)  
UNITS "seconds"  
MAX-ACCESS read-write  
STATUS current  
DESCRIPTION

"The default ICMP idle timeout parameter.

If the agent is capable of storing non-volatile configuration, then the value of this object must be restored after a re-initialization of the management system."

DEFVAL { 300 }  
::= { natDefTimeouts 3 }

```
--
-- Other protocol parameters
--

natOtherDefIdleTimeout OBJECT-TYPE
    SYNTAX      Unsigned32  (1..4294967295)
    UNITS       "seconds"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The default idle timeout parameter for protocols
        represented by the value other (2) in
        NatProtocolType.

        If the agent is capable of storing non-volatile
        configuration, then the value of this object must be
        restored after a re-initialization of the management
        system."
    DEFVAL { 60 }
    ::= { natDefTimeouts 4 }

--
-- TCP related NAT Timers
--

natTcpDefIdleTimeout OBJECT-TYPE
    SYNTAX      Unsigned32  (1..4294967295)
    UNITS       "seconds"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The default time interval that a NAT session for an
        established TCP connection is allowed to remain
        valid without any activity on the TCP connection.

        If the agent is capable of storing non-volatile
        configuration, then the value of this object must be
        restored after a re-initialization of the management
        system."
    DEFVAL { 86400 }
    ::= { natDefTimeouts 5 }

natTcpDefNegTimeout OBJECT-TYPE
    SYNTAX      Unsigned32  (1..4294967295)
    UNITS       "seconds"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
```

"The default time interval that a NAT session for a TCP connection that is not in the established state is allowed to remain valid without any activity on the TCP connection.

If the agent is capable of storing non-volatile configuration, then the value of this object must be restored after a re-initialization of the management system."

```
DEFVAL { 60 }
::= { natDefTimeouts 6 }
```

#### natNotifThrottlingInterval OBJECT-TYPE

SYNTAX Integer32 (0 | 5..3600)

UNITS "seconds"

MAX-ACCESS read-write

STATUS current

#### DESCRIPTION

"This object controls the generation of the natPacketDiscard notification.

If this object has a value of zero, then no natPacketDiscard notifications will be transmitted by the agent.

If this object has a non-zero value, then the agent must not generate more than one natPacketDiscard 'notification-event' in the indicated period, where a 'notification-event' is the generation of a single notification PDU type to a list of notification destinations. If additional NAT packets are discarded within the throttling period, then notification-events for these changes must be suppressed by the agent until the current throttling period expires.

If natNotifThrottlingInterval notification generation is enabled, the suggested default throttling period is 60 seconds, but generation of the natPacketDiscard notification should be disabled by default.

If the agent is capable of storing non-volatile configuration, then the value of this object must be restored after a re-initialization of the management system.

The actual transmission of notifications is controlled via the MIB modules in RFC 3413."

```
DEFVAL { 0 }
```

```

 ::= { natNotifCtrl 1 }

--
-- The NAT Interface Table
--

natInterfaceTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF NatInterfaceEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table specifies the attributes for interfaces on a
         device supporting NAT function."
    ::= { natMIBObjects 3 }

natInterfaceEntry OBJECT-TYPE
    SYNTAX      NatInterfaceEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Each entry in the natInterfaceTable holds a set of
         parameters for an interface, instantiated by
         ifIndex. Therefore, the interface index must have been
         assigned, according to the applicable procedures,
         before it can be meaningfully used.
         Generally, this means that the interface must exist.

         When natStorageType is of type nonVolatile, however,
         this may reflect the configuration for an interface whose
         ifIndex has been assigned but for which the supporting
         implementation is not currently present."
    INDEX      { ifIndex }
    ::= { natInterfaceTable 1 }

NatInterfaceEntry ::= SEQUENCE {
    natInterfaceRealm          INTEGER,
    natInterfaceServiceType    BITS,
    natInterfaceInTranslates   Counter64,
    natInterfaceOutTranslates  Counter64,
    natInterfaceDiscards       Counter64,
    natInterfaceStorageType    StorageType,
    natInterfaceRowStatus      RowStatus
}

natInterfaceRealm OBJECT-TYPE
    SYNTAX      INTEGER {
        private (1),
        public (2)
    }

```

```
    }
    MAX-ACCESS read-create
    STATUS      current
    DESCRIPTION
        "This object identifies whether this interface is
        connected to the private or the public realm."
    DEFVAL { public }
    ::= { natInterfaceEntry 1 }
```

natInterfaceServiceType OBJECT-TYPE

```
    SYNTAX      BITS {
        basicNat (0),
        napt (1),
        bidirectionalNat (2),
        twiceNat (3)
    }
    MAX-ACCESS read-create
    STATUS      current
    DESCRIPTION
        "An indication of the direction in which new sessions
        are permitted and the extent of translation done within
        the IP and transport headers."
    ::= { natInterfaceEntry 2 }
```

natInterfaceInTranslates OBJECT-TYPE

```
    SYNTAX      Counter64
    MAX-ACCESS read-only
    STATUS      current
    DESCRIPTION
        "Number of packets received on this interface that
        were translated.
        Discontinuities in the value of this counter can occur at
        reinitialization of the management system and at other
        times as indicated by the value of
        ifCounterDiscontinuityTime on the relevant interface."
    ::= { natInterfaceEntry 3 }
```

natInterfaceOutTranslates OBJECT-TYPE

```
    SYNTAX      Counter64
    MAX-ACCESS read-only
    STATUS      current
    DESCRIPTION
        "Number of translated packets that were sent out this
        interface.

        Discontinuities in the value of this counter can occur at
        reinitialization of the management system and at other
        times as indicated by the value of
```

```
        ifCounterDiscontinuityTime on the relevant interface."
 ::= { natInterfaceEntry 4 }
```

natInterfaceDiscards OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Number of packets that had to be rejected/dropped due to a lack of resources for this interface.

Discontinuities in the value of this counter can occur at reinitialization of the management system and at other times as indicated by the value of

```
        ifCounterDiscontinuityTime on the relevant interface."
 ::= { natInterfaceEntry 5 }
```

natInterfaceStorageType OBJECT-TYPE

SYNTAX StorageType

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The storage type for this conceptual row. Conceptual rows having the value 'permanent' need not allow write-access to any columnar objects in the row."

REFERENCE

"Textual Conventions for SMIV2, Section 2."

DEFVAL { nonVolatile }

```
 ::= { natInterfaceEntry 6 }
```

natInterfaceRowStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The status of this conceptual row.

Until instances of all corresponding columns are appropriately configured, the value of the corresponding instance of the natInterfaceRowStatus column is 'notReady'.

In particular, a newly created row cannot be made active until the corresponding instance of natInterfaceServiceType has been set.

None of the objects in this row may be modified while the value of this object is active(1)."

## REFERENCE

"Textual Conventions for SMIV2, Section 2."

::= { natInterfaceEntry 7 }

--

-- The Address Map Table

--

natAddrMapTable OBJECT-TYPE

SYNTAX SEQUENCE OF NatAddrMapEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table lists address map parameters for NAT."

::= { natMIBObjects 4 }

natAddrMapEntry OBJECT-TYPE

SYNTAX NatAddrMapEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This entry represents an address map to be used for NAT and contributes to the dynamic and/or static address mapping tables of the NAT device."

INDEX { ifIndex, natAddrMapIndex }

::= { natAddrMapTable 1 }

NatAddrMapEntry ::= SEQUENCE {

natAddrMapIndex

natAddrMapName

natAddrMapEntryType

natAddrMapTranslationEntity

natAddrMapLocalAddrType

natAddrMapLocalAddrFrom

natAddrMapLocalAddrTo

natAddrMapLocalPortFrom

natAddrMapLocalPortTo

natAddrMapGlobalAddrType

natAddrMapGlobalAddrFrom

natAddrMapGlobalAddrTo

natAddrMapGlobalPortFrom

natAddrMapGlobalPortTo

natAddrMapProtocol

natAddrMapInTranslates

natAddrMapOutTranslates

natAddrMapDiscards

NatAddrMapId,

SnmpAdminString,

NatAssociationType,

NatTranslationEntity,

InetAddressType,

InetAddress,

InetAddress,

InetPortNumber,

InetPortNumber,

InetAddressType,

InetAddress,

InetAddress,

InetPortNumber,

InetPortNumber,

NatProtocolMap,

Counter64,

Counter64,

Counter64,

```

    natAddrMapAddrUsed          Gauge32,
    natAddrMapStorageType      StorageType,
    natAddrMapRowStatus        RowStatus
}

natAddrMapIndex OBJECT-TYPE
    SYNTAX      NatAddrMapId
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Along with ifIndex, this object uniquely
         identifies an entry in the natAddrMapTable.
         Address map entries are applied in the order
         specified by natAddrMapIndex."
    ::= { natAddrMapEntry 1 }

natAddrMapName OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE(1..32))
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Name identifying all map entries in the table associated
         with the same interface. All map entries with the same
         ifIndex MUST have the same map name."
    ::= { natAddrMapEntry 2 }

natAddrMapEntryType OBJECT-TYPE
    SYNTAX      NatAssociationType
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This parameter can be used to set up static
         or dynamic address maps."
    ::= { natAddrMapEntry 3 }

natAddrMapTranslationEntity OBJECT-TYPE
    SYNTAX      NatTranslationEntity
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The end-point entity (source or destination) in
         inbound or outbound sessions (i.e., first packets) that
         may be translated by an address map entry.

         Session direction (inbound or outbound) is
         derived from the direction of the first packet
         of a session traversing a NAT interface.
         NAT address (and Transport-ID) maps may be defined

```

to effect inbound or outbound sessions.

Traditionally, address maps for Basic NAT and NAPT are configured on a public interface for outbound sessions, effecting translation of source end-point. The value of this object must be set to `outboundSrcEndPoint` for those interfaces.

Alternately, if address maps for Basic NAT and NAPT were to be configured on a private interface, the desired value for this object for the map entries would be `inboundSrcEndPoint` (i.e., effecting translation of source end-point for inbound sessions).

If `TwiceNAT` were to be configured on a private interface, the desired value for this object for the map entries would be a bitmask of `inboundSrcEndPoint` and `inboundDstEndPoint`."

```
::= { natAddrMapEntry 4 }
```

`natAddrMapLocalAddrType` OBJECT-TYPE

SYNTAX `InetAddressType`

MAX-ACCESS `read-create`

STATUS `current`

DESCRIPTION

"This object specifies the address type used for `natAddrMapLocalAddrFrom` and `natAddrMapLocalAddrTo`."

```
::= { natAddrMapEntry 5 }
```

`natAddrMapLocalAddrFrom` OBJECT-TYPE

SYNTAX `InetAddress`

MAX-ACCESS `read-create`

STATUS `current`

DESCRIPTION

"This object specifies the first IP address of the range of IP addresses mapped by this translation entry. The value of this object must be less than or equal to the value of the `natAddrMapLocalAddrTo` object.

The type of this address is determined by the value of the `natAddrMapLocalAddrType` object."

```
::= { natAddrMapEntry 6 }
```

`natAddrMapLocalAddrTo` OBJECT-TYPE

SYNTAX `InetAddress`

MAX-ACCESS `read-create`

STATUS `current`

DESCRIPTION

"This object specifies the last IP address of the range of IP addresses mapped by this translation entry. If only a single address is being mapped, the value of this object is equal to the value of natAddrMapLocalAddrFrom. For a static NAT, the number of addresses in the range defined by natAddrMapLocalAddrFrom and natAddrMapLocalAddrTo must be equal to the number of addresses in the range defined by natAddrMapGlobalAddrFrom and natAddrMapGlobalAddrTo. The value of this object must be greater than or equal to the value of the natAddrMapLocalAddrFrom object.

The type of this address is determined by the value of the natAddrMapLocalAddrType object."

::= { natAddrMapEntry 7 }

#### natAddrMapLocalPortFrom OBJECT-TYPE

SYNTAX InetPortNumber

MAX-ACCESS read-create

STATUS current

#### DESCRIPTION

"If this conceptual row describes a Basic NAT address mapping, then the value of this object must be zero. If this conceptual row describes NAPT, then the value of this object specifies the first port number in the range of ports being mapped.

The value of this object must be less than or equal to the value of the natAddrMapLocalPortTo object. If the translation specifies a single port, then the value of this object is equal to the value of natAddrMapLocalPortTo."

DEFVAL { 0 }

::= { natAddrMapEntry 8 }

#### natAddrMapLocalPortTo OBJECT-TYPE

SYNTAX InetPortNumber

MAX-ACCESS read-create

STATUS current

#### DESCRIPTION

"If this conceptual row describes a Basic NAT address mapping, then the value of this object must be zero. If this conceptual row describes NAPT, then the value of this object specifies the last port number in the range of ports being mapped.

The value of this object must be greater than or equal to the value of the natAddrMapLocalPortFrom object. If the translation specifies a single port, then the value of this object is equal to the value of natAddrMapLocalPortFrom."

```
DEFVAL { 0 }  
::= { natAddrMapEntry 9 }
```

natAddrMapGlobalAddrType OBJECT-TYPE

```
SYNTAX      InetAddressType  
MAX-ACCESS  read-create  
STATUS      current  
DESCRIPTION  
    "This object specifies the address type used for  
    natAddrMapGlobalAddrFrom and natAddrMapGlobalAddrTo."  
::= { natAddrMapEntry 10 }
```

natAddrMapGlobalAddrFrom OBJECT-TYPE

```
SYNTAX      InetAddress  
MAX-ACCESS  read-create  
STATUS      current  
DESCRIPTION  
    "This object specifies the first IP address of the range of  
    IP addresses being mapped to. The value of this object  
    must be less than or equal to the value of the  
    natAddrMapGlobalAddrTo object.  
  
    The type of this address is determined by the value of  
    the natAddrMapGlobalAddrType object."  
::= { natAddrMapEntry 11 }
```

natAddrMapGlobalAddrTo OBJECT-TYPE

```
SYNTAX      InetAddress  
MAX-ACCESS  read-create  
STATUS      current  
DESCRIPTION  
    "This object specifies the last IP address of the range of  
    IP addresses being mapped to. If only a single address is  
    being mapped to, the value of this object is equal to the  
    value of natAddrMapGlobalAddrFrom. For a static NAT, the  
    number of addresses in the range defined by  
    natAddrMapGlobalAddrFrom and natAddrMapGlobalAddrTo must be  
    equal to the number of addresses in the range defined by  
    natAddrMapLocalAddrFrom and natAddrMapLocalAddrTo.  
    The value of this object must be greater than or equal to  
    the value of the natAddrMapGlobalAddrFrom object.  
  
    The type of this address is determined by the value of  
    the natAddrMapGlobalAddrType object."  
::= { natAddrMapEntry 12 }
```

natAddrMapGlobalPortFrom OBJECT-TYPE

```
SYNTAX      InetPortNumber
```

MAX-ACCESS read-create  
STATUS current  
DESCRIPTION

"If this conceptual row describes a Basic NAT address mapping, then the value of this object must be zero. If this conceptual row describes NAPT, then the value of this object specifies the first port number in the range of ports being mapped to.

The value of this object must be less than or equal to the value of the natAddrMapGlobalPortTo object. If the translation specifies a single port, then the value of this object is equal to the value natAddrMapGlobalPortTo."

DEFVAL { 0 }  
::= { natAddrMapEntry 13 }

natAddrMapGlobalPortTo OBJECT-TYPE

SYNTAX InetPortNumber  
MAX-ACCESS read-create  
STATUS current  
DESCRIPTION

"If this conceptual row describes a Basic NAT address mapping, then the value of this object must be zero. If this conceptual row describes NAPT, then the value of this object specifies the last port number in the range of ports being mapped to.

The value of this object must be greater than or equal to the value of the natAddrMapGlobalPortFrom object. If the translation specifies a single port, then the value of this object is equal to the value of natAddrMapGlobalPortFrom."

DEFVAL { 0 }  
::= { natAddrMapEntry 14 }

natAddrMapProtocol OBJECT-TYPE

SYNTAX NatProtocolMap  
MAX-ACCESS read-create  
STATUS current  
DESCRIPTION

"This object specifies a bitmap of protocol identifiers."

::= { natAddrMapEntry 15 }

natAddrMapInTranslates OBJECT-TYPE

SYNTAX Counter64  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION

"The number of inbound packets pertaining to this address map entry that were translated.

Discontinuities in the value of this counter can occur at reinitialization of the management system and at other times, as indicated by the value of ifCounterDiscontinuityTime on the relevant interface."

::= { natAddrMapEntry 16 }

natAddrMapOutTranslates OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of outbound packets pertaining to this address map entry that were translated.

Discontinuities in the value of this counter can occur at reinitialization of the management system and at other times, as indicated by the value of ifCounterDiscontinuityTime on the relevant interface."

::= { natAddrMapEntry 17 }

natAddrMapDiscards OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of packets pertaining to this address map entry that were dropped due to lack of addresses in the address pool identified by this address map. The value of this object must always be zero in case of static address map.

Discontinuities in the value of this counter can occur at reinitialization of the management system and at other times, as indicated by the value of ifCounterDiscontinuityTime on the relevant interface."

::= { natAddrMapEntry 18 }

natAddrMapAddrUsed OBJECT-TYPE

SYNTAX Gauge32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of addresses pertaining to this address map that are currently being used from the NAT pool.

The value of this object must always be zero in the case

of a static address map."  
 ::= { natAddrMapEntry 19 }

#### natAddrMapStorageType OBJECT-TYPE

SYNTAX       StorageType  
 MAX-ACCESS   read-create  
 STATUS       current  
 DESCRIPTION  
     "The storage type for this conceptual row.  
     Conceptual rows having the value 'permanent'  
     need not allow write-access to any columnar objects  
     in the row."  
 REFERENCE  
     "Textual Conventions for SMIV2, Section 2."  
 DEFVAL { nonVolatile }  
 ::= { natAddrMapEntry 20 }

#### natAddrMapRowStatus OBJECT-TYPE

SYNTAX       RowStatus  
 MAX-ACCESS   read-create  
 STATUS       current  
 DESCRIPTION  
     "The status of this conceptual row.  
  
     Until instances of all corresponding columns are  
     appropriately configured, the value of the  
     corresponding instance of the natAddrMapRowStatus  
     column is 'notReady'.  
  
     None of the objects in this row may be modified  
     while the value of this object is active(1)."  
 REFERENCE  
     "Textual Conventions for SMIV2, Section 2."  
 ::= { natAddrMapEntry 21 }

--  
 -- Address Bind section  
 --

#### natAddrBindNumberOfEntries OBJECT-TYPE

SYNTAX       Gauge32  
 MAX-ACCESS   read-only  
 STATUS       current  
 DESCRIPTION  
     "This object maintains a count of the number of entries  
     that currently exist in the natAddrBindTable."  
 ::= { natMIBObjects 5 }

```

--
-- The NAT Address BIND Table
--

natAddrBindTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF NatAddrBindEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table holds information about the currently
        active NAT BINDs."
    ::= { natMIBObjects 6 }

natAddrBindEntry OBJECT-TYPE
    SYNTAX      NatAddrBindEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Each entry in this table holds information about
        an active address BIND.  These entries are lost
        upon agent restart.

        This row has indexing which may create variables with
        more than 128 subidentifiers.  Implementers of this table
        must be careful not to create entries that would result
        in OIDs which exceed the 128 subidentifier limit.
        Otherwise, the information cannot be accessed using
        SNMPv1, SNMPv2c or SNMPv3."

    INDEX      { ifIndex, natAddrBindLocalAddrType, natAddrBindLocalAddr }
    ::= { natAddrBindTable 1 }

NatAddrBindEntry ::= SEQUENCE {
    natAddrBindLocalAddrType      InetAddressType,
    natAddrBindLocalAddr          InetAddress,
    natAddrBindGlobalAddrType     InetAddressType,
    natAddrBindGlobalAddr         InetAddress,
    natAddrBindId                 NatBindId,
    natAddrBindTranslationEntity  NatTranslationEntity,
    natAddrBindType               NatAssociationType,
    natAddrBindMapIndex           NatAddrMapId,
    natAddrBindSessions           Gauge32,
    natAddrBindMaxIdleTime        TimeTicks,
    natAddrBindCurrentIdleTime    TimeTicks,
    natAddrBindInTranslates       Counter64,
    natAddrBindOutTranslates      Counter64
}

```

**natAddrBindLocalAddrType OBJECT-TYPE**

SYNTAX InetAddressType

MAX-ACCESS not-accessible

STATUS current

**DESCRIPTION**

"This object specifies the address type used for  
natAddrBindLocalAddr."

::= { natAddrBindEntry 1 }

**natAddrBindLocalAddr OBJECT-TYPE**

SYNTAX InetAddress

MAX-ACCESS not-accessible

STATUS current

**DESCRIPTION**

"This object represents the private-realm specific network  
layer address, which maps to the public-realm address  
represented by natAddrBindGlobalAddr."

The type of this address is determined by the value of  
the natAddrBindLocalAddrType object."

::= { natAddrBindEntry 2 }

**natAddrBindGlobalAddrType OBJECT-TYPE**

SYNTAX InetAddressType

MAX-ACCESS read-only

STATUS current

**DESCRIPTION**

"This object specifies the address type used for  
natAddrBindGlobalAddr."

::= { natAddrBindEntry 3 }

**natAddrBindGlobalAddr OBJECT-TYPE**

SYNTAX InetAddress

MAX-ACCESS read-only

STATUS current

**DESCRIPTION**

"This object represents the public-realm network layer  
address that maps to the private-realm network layer  
address represented by natAddrBindLocalAddr."

The type of this address is determined by the value of  
the natAddrBindGlobalAddrType object."

::= { natAddrBindEntry 4 }

**natAddrBindId OBJECT-TYPE**

SYNTAX NatBindId

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"This object represents a bind id that is dynamically assigned to each bind by a NAT enabled device. Each bind is represented by a bind id that is unique across both, the natAddrBindTable and the natAddrPortBindTable."

::= { natAddrBindEntry 5 }

## natAddrBindTranslationEntity OBJECT-TYPE

SYNTAX NatTranslationEntity

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"This object represents the direction of sessions for which this bind is applicable and the endpoint entity (source or destination) within the sessions that is subject to translation using the BIND.

Orientation of the bind can be a superset of translationEntity of the address map entry which forms the basis for this bind.

For example, if the translationEntity of an address map entry is outboundSrcEndPoint, the translationEntity of a bind derived from this map entry may either be outboundSrcEndPoint or it may be bidirectional (a bitmask of outboundSrcEndPoint and inboundDstEndPoint)."

::= { natAddrBindEntry 6 }

## natAddrBindType OBJECT-TYPE

SYNTAX NatAssociationType

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"This object indicates whether the bind is static or dynamic."

::= { natAddrBindEntry 7 }

## natAddrBindMapIndex OBJECT-TYPE

SYNTAX NatAddrMapId

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"This object is a pointer to the natAddrMapTable entry (and the parameters of that entry) which was used in creating this BIND. This object, in conjunction with the ifIndex (which identifies a unique addrMapName) points to

a unique entry in the natAddrMapTable."  
 ::= { natAddrBindEntry 8 }

natAddrBindSessions OBJECT-TYPE

SYNTAX Gauge32  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
 "Number of sessions currently using this BIND."  
 ::= { natAddrBindEntry 9 }

natAddrBindMaxIdleTime OBJECT-TYPE

SYNTAX TimeTicks  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
 "This object indicates the maximum time for  
 which this bind can be idle with no sessions  
 attached to it.  
  
 The value of this object is of relevance only for  
 dynamic NAT."  
 ::= { natAddrBindEntry 10 }

natAddrBindCurrentIdleTime OBJECT-TYPE

SYNTAX TimeTicks  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
 "At any given instance, this object indicates the  
 time that this bind has been idle without any sessions  
 attached to it.  
  
 The value of this object is of relevance only for  
 dynamic NAT."  
 ::= { natAddrBindEntry 11 }

natAddrBindInTranslates OBJECT-TYPE

SYNTAX Counter64  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
 "The number of inbound packets that were successfully  
 translated by using this bind entry.  
  
 Discontinuities in the value of this counter can occur at  
 reinitialization of the management system and at other  
 times, as indicated by the value of

```

        ifCounterDiscontinuityTime on the relevant interface."
 ::= { natAddrBindEntry 12 }

natAddrBindOutTranslates OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of outbound packets that were successfully
        translated using this bind entry.

        Discontinuities in the value of this counter can occur at
        reinitialization of the management system and at other
        times as indicated by the value of
        ifCounterDiscontinuityTime on the relevant interface."
 ::= { natAddrBindEntry 13 }

--
-- Address Port Bind section
--

natAddrPortBindNumberOfEntries OBJECT-TYPE
    SYNTAX      Gauge32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object maintains a count of the number of entries
        that currently exist in the natAddrPortBindTable."
 ::= { natMIBObjects 7 }

--
-- The NAT Address Port Bind Table
--

natAddrPortBindTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF NatAddrPortBindEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table holds information about the currently
        active NAT BINDs."
 ::= { natMIBObjects 8 }

natAddrPortBindEntry OBJECT-TYPE
    SYNTAX      NatAddrPortBindEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION

```

"Each entry in the this table holds information about a NATP bind that is currently active. These entries are lost upon agent restart.

This row has indexing which may create variables with more than 128 subidentifiers. Implementers of this table must be careful not to create entries which would result in OIDs that exceed the 128 subidentifier limit. Otherwise, the information cannot be accessed using SNMPv1, SNMPv2c or SNMPv3."

```
INDEX { ifIndex, natAddrPortBindLocalAddrType,
        natAddrPortBindLocalAddr, natAddrPortBindLocalPort,
        natAddrPortBindProtocol }
 ::= { natAddrPortBindTable 1 }
```

```
NatAddrPortBindEntry ::= SEQUENCE {
    natAddrPortBindLocalAddrType      InetAddressType,
    natAddrPortBindLocalAddr          InetAddress,
    natAddrPortBindLocalPort          InetPortNumber,
    natAddrPortBindProtocol            NatProtocolType,
    natAddrPortBindGlobalAddrType      InetAddressType,
    natAddrPortBindGlobalAddr          InetAddress,
    natAddrPortBindGlobalPort          InetPortNumber,
    natAddrPortBindId                  NatBindId,
    natAddrPortBindTranslationEntity   NatTranslationEntity,
    natAddrPortBindType                 NatAssociationType,
    natAddrPortBindMapIndex             NatAddrMapId,
    natAddrPortBindSessions             Gauge32,
    natAddrPortBindMaxIdleTime           TimeTicks,
    natAddrPortBindCurrentIdleTime       TimeTicks,
    natAddrPortBindInTranslates          Counter64,
    natAddrPortBindOutTranslates         Counter64
}
```

natAddrPortBindLocalAddrType OBJECT-TYPE

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

"This object specifies the address type used for natAddrPortBindLocalAddr."

```
::= { natAddrPortBindEntry 1 }
```

natAddrPortBindLocalAddr OBJECT-TYPE

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

"This object represents the private-realm specific network layer address which, in conjunction with natAddrPortBindLocalPort, maps to the public-realm network layer address and transport id represented by natAddrPortBindGlobalAddr and natAddrPortBindGlobalPort respectively.

The type of this address is determined by the value of the natAddrPortBindLocalAddrType object."

::= { natAddrPortBindEntry 2 }

#### natAddrPortBindLocalPort OBJECT-TYPE

SYNTAX InetPortNumber

MAX-ACCESS not-accessible

STATUS current

#### DESCRIPTION

"For a protocol value TCP or UDP, this object represents the private-realm specific port number. On the other hand, for ICMP a bind is created only for query/response type ICMP messages such as ICMP echo, Timestamp, and Information request messages, and this object represents the private-realm specific identifier in the ICMP message, as defined in RFC 792 for ICMPv4 and in RFC 2463 for ICMPv6.

This object, together with natAddrPortBindProtocol, natAddrPortBindLocalAddrType, and natAddrPortBindLocalAddr, constitutes a session endpoint in the private realm. A bind entry binds a private realm specific endpoint to a public realm specific endpoint, as represented by the tuple of (natAddrPortBindGlobalPort, natAddrPortBindProtocol, natAddrPortBindGlobalAddrType, and natAddrPortBindGlobalAddr)."

::= { natAddrPortBindEntry 3 }

#### natAddrPortBindProtocol OBJECT-TYPE

SYNTAX NatProtocolType

MAX-ACCESS not-accessible

STATUS current

#### DESCRIPTION

"This object specifies a protocol identifier. If the value of this object is none(1), then this bind entry applies to all IP traffic. Any other value of this object specifies the class of IP traffic to which this BIND applies."

::= { natAddrPortBindEntry 4 }

**natAddrPortBindGlobalAddrType OBJECT-TYPE**

SYNTAX InetAddressType

MAX-ACCESS read-only

STATUS current

**DESCRIPTION**

"This object specifies the address type used for  
natAddrPortBindGlobalAddr."

::= { natAddrPortBindEntry 5 }

**natAddrPortBindGlobalAddr OBJECT-TYPE**

SYNTAX InetAddress

MAX-ACCESS read-only

STATUS current

**DESCRIPTION**

"This object represents the public-realm specific network  
layer address that, in conjunction with  
natAddrPortBindGlobalPort, maps to the private-realm

network layer address and transport id represented by  
natAddrPortBindLocalAddr and natAddrPortBindLocalPort,  
respectively.

The type of this address is determined by the value of  
the natAddrPortBindGlobalAddrType object."

::= { natAddrPortBindEntry 6 }

**natAddrPortBindGlobalPort OBJECT-TYPE**

SYNTAX InetPortNumber

MAX-ACCESS read-only

STATUS current

**DESCRIPTION**

"For a protocol value TCP or UDP, this object represents  
the public-realm specific port number. On the other  
hand, for ICMP a bind is created only for query/response  
type ICMP messages such as ICMP echo, Timestamp, and  
Information request messages, and this object represents  
the public-realm specific identifier in the ICMP message,  
as defined in RFC 792 for ICMPv4 and in RFC 2463 for  
ICMPv6.

This object, together with natAddrPortBindProtocol,  
natAddrPortBindGlobalAddrType, and  
natAddrPortBindGlobalAddr, constitutes a session endpoint  
in the public realm. A bind entry binds a public realm  
specific endpoint to a private realm specific endpoint,  
as represented by the tuple of  
(natAddrPortBindLocalPort, natAddrPortBindProtocol,  
natAddrPortBindLocalAddrType, and

```
        natAddrPortBindLocalAddr)."
 ::= { natAddrPortBindEntry 7 }
```

natAddrPortBindId OBJECT-TYPE

SYNTAX NatBindId

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object represents a bind id that is dynamically assigned to each bind by a NAT enabled device. Each bind is represented by a unique bind id across both the natAddrBindTable and the natAddrPortBindTable."

```
 ::= { natAddrPortBindEntry 8 }
```

natAddrPortBindTranslationEntity OBJECT-TYPE

SYNTAX NatTranslationEntity

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object represents the direction of sessions for which this bind is applicable and the entity (source or destination) within the sessions that is subject to translation with the BIND.

Orientation of the bind can be a superset of the translationEntity of the address map entry that forms the basis for this bind.

For example, if the translationEntity of an address map entry is outboundSrcEndPoint, the translationEntity of a bind derived from this map entry may either be outboundSrcEndPoint or may be bidirectional (a bitmask of outboundSrcEndPoint and inboundDstEndPoint)."

```
 ::= { natAddrPortBindEntry 9 }
```

natAddrPortBindType OBJECT-TYPE

SYNTAX NatAssociationType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object indicates whether the bind is static or dynamic."

```
 ::= { natAddrPortBindEntry 10 }
```

natAddrPortBindMapIndex OBJECT-TYPE

SYNTAX NatAddrMapId

MAX-ACCESS read-only

STATUS current  
DESCRIPTION  
    "This object is a pointer to the natAddrMapTable entry  
    (and the parameters of that entry) used in  
    creating this BIND. This object, in conjunction with the  
    ifIndex (which identifies a unique addrMapName), points  
    to a unique entry in the natAddrMapTable."  
 ::= { natAddrPortBindEntry 11 }

natAddrPortBindSessions OBJECT-TYPE

SYNTAX Gauge32  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
    "Number of sessions currently using this BIND."  
 ::= { natAddrPortBindEntry 12 }

natAddrPortBindMaxIdleTime OBJECT-TYPE

SYNTAX TimeTicks  
MAX-ACCESS read-only  
STATUS current  
  
DESCRIPTION  
    "This object indicates the maximum time for  
    which this bind can be idle without any sessions  
    attached to it.  
    The value of this object is of relevance  
    only for dynamic NAT."  
 ::= { natAddrPortBindEntry 13 }

natAddrPortBindCurrentIdleTime OBJECT-TYPE

SYNTAX TimeTicks  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
    "At any given instance, this object indicates the  
    time that this bind has been idle without any sessions  
    attached to it.  
  
    The value of this object is of relevance  
    only for dynamic NAT."  
 ::= { natAddrPortBindEntry 14 }

natAddrPortBindInTranslates OBJECT-TYPE

SYNTAX Counter64  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION

"The number of inbound packets that were translated as per this bind entry.

Discontinuities in the value of this counter can occur at reinitialization of the management system and at other times, as indicated by the value of ifCounterDiscontinuityTime on the relevant interface."

::= { natAddrPortBindEntry 15 }

natAddrPortBindOutTranslates OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of outbound packets that were translated as per this bind entry.

Discontinuities in the value of this counter can occur at reinitialization of the management system and at other times, as indicated by the value of ifCounterDiscontinuityTime on the relevant interface."

::= { natAddrPortBindEntry 16 }

--

-- The Session Table

--

natSessionTable OBJECT-TYPE

SYNTAX SEQUENCE OF NatSessionEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The (conceptual) table containing one entry for each NAT session currently active on this NAT device."

::= { natMIBObjects 9 }

natSessionEntry OBJECT-TYPE

SYNTAX NatSessionEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry (conceptual row) containing information about an active NAT session on this NAT device. These entries are lost upon agent restart."

INDEX { ifIndex, natSessionIndex }

::= { natSessionTable 1 }

NatSessionEntry ::= SEQUENCE {

|                                |                  |
|--------------------------------|------------------|
| natSessionIndex                | NatSessionId,    |
| natSessionPrivateSrcEPBindId   | NatBindIdOrZero, |
| natSessionPrivateSrcEPBindMode | NatBindMode,     |
| natSessionPrivateDstEPBindId   | NatBindIdOrZero, |
| natSessionPrivateDstEPBindMode | NatBindMode,     |
| natSessionDirection            | INTEGER,         |
| natSessionUpTime               | TimeTicks,       |
| natSessionAddrMapIndex         | NatAddrMapId,    |
| natSessionProtocolType         | NatProtocolType, |
| natSessionPrivateAddrType      | InetAddressType, |
| natSessionPrivateSrcAddr       | InetAddress,     |
| natSessionPrivateSrcPort       | InetPortNumber,  |
| natSessionPrivateDstAddr       | InetAddress,     |
| natSessionPrivateDstPort       | InetPortNumber,  |
| natSessionPublicAddrType       | InetAddressType, |
| natSessionPublicSrcAddr        | InetAddress,     |
| natSessionPublicSrcPort        | InetPortNumber,  |
| natSessionPublicDstAddr        | InetAddress,     |
| natSessionPublicDstPort        | InetPortNumber,  |
| natSessionMaxIdleTime          | TimeTicks,       |
| natSessionCurrentIdleTime      | TimeTicks,       |
| natSessionInTranslates         | Counter64,       |
| natSessionOutTranslates        | Counter64        |

}

natSessionIndex OBJECT-TYPE  
 SYNTAX NatSessionId  
 MAX-ACCESS not-accessible  
 STATUS current  
 DESCRIPTION  
     "The session ID for this NAT session."  
 ::= { natSessionEntry 1 }

natSessionPrivateSrcEPBindId OBJECT-TYPE  
 SYNTAX NatBindIdOrZero  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION  
     "The bind id associated between private and public  
     source end points. In the case of Symmetric-NAT,  
     this should be set to zero."  
 ::= { natSessionEntry 2 }

natSessionPrivateSrcEPBindMode OBJECT-TYPE  
 SYNTAX NatBindMode  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION

```
        "This object indicates whether the bind indicated
        by the object natSessionPrivateSrcEPBindId
        is an address bind or an address port bind."
 ::= { natSessionEntry 3 }
```

natSessionPrivateDstePBindId OBJECT-TYPE

```
SYNTAX      NatBindIdOrZero
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The bind id associated between private and public
    destination end points."
 ::= { natSessionEntry 4 }
```

natSessionPrivateDstePBindMode OBJECT-TYPE

```
SYNTAX      NatBindMode
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "This object indicates whether the bind indicated
    by the object natSessionPrivateDstePBindId
    is an address bind or an address port bind."
 ::= { natSessionEntry 5 }
```

natSessionDirection OBJECT-TYPE

```
SYNTAX      INTEGER {
                inbound (1),
                outbound (2)
            }
```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The direction of this session with respect to the local network. 'inbound' indicates that this session was initiated from the public network into the private network. 'outbound' indicates that this session was initiated from the private network into the public network."

```
 ::= { natSessionEntry 6 }
```

natSessionUpTime OBJECT-TYPE

SYNTAX TimeTicks

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The up time of this session in one-hundredths of a second."

```
::= { natSessionEntry 7 }
```

natSessionAddrMapIndex OBJECT-TYPE

SYNTAX NatAddrMapId

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object is a pointer to the natAddrMapTable entry (and the parameters of that entry) used in creating this session. This object, in conjunction with the ifIndex (which identifies a unique addrMapName), points to a unique entry in the natAddrMapTable."

```
::= { natSessionEntry 8 }
```

natSessionProtocolType OBJECT-TYPE

SYNTAX NatProtocolType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The protocol type of this session."

```
::= { natSessionEntry 9 }
```

natSessionPrivateAddrType OBJECT-TYPE

SYNTAX InetAddressType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object specifies the address type used for natSessionPrivateSrcAddr and natSessionPrivateDstAddr."

```
::= { natSessionEntry 10 }
```

natSessionPrivateSrcAddr OBJECT-TYPE

SYNTAX InetAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The source IP address of the session endpoint that lies in the private network."

The value of this object must be zero only when the natSessionPrivateSrcEPBindId object has a zero value. When the value of this object is zero, the NAT session lookup will match any IP address to this field.

The type of this address is determined by the value of the natSessionPrivateAddrType object."

```
::= { natSessionEntry 11 }
```

**natSessionPrivateSrcPort OBJECT-TYPE**

SYNTAX       InetPortNumber

MAX-ACCESS   read-only

STATUS       current

**DESCRIPTION**

"When the value of protocol is TCP or UDP, this object represents the source port in the first packet of session while in private-realm. On the other hand, when the protocol is ICMP, a NAT session is created only for query/response type ICMP messages such as ICMP echo, Timestamp, and Information request messages, and this object represents the private-realm specific identifier in the ICMP message, as defined in RFC 792 for ICMPv4 and in RFC 2463 for ICMPv6.

The value of this object must be zero when the natSessionPrivateSrcEPBindId object has zero value and value of natSessionPrivateSrcEPBindMode is addressPortBind(2). In such a case, the NAT session lookup will match any port number to this field.

The value of this object must be zero when the object is not a representative field (SrcPort, DstPort, or ICMP identifier) of the session tuple in either the public realm or the private realm."

::= { natSessionEntry 12 }

**natSessionPrivateDstAddr OBJECT-TYPE**

SYNTAX       InetAddress

MAX-ACCESS   read-only

STATUS       current

**DESCRIPTION**

"The destination IP address of the session endpoint that lies in the private network.

The value of this object must be zero when the natSessionPrivateDstEPBindId object has a zero value. In such a scenario, the NAT session lookup will match any IP address to this field.

The type of this address is determined by the value of the natSessionPrivateAddrType object."

::= { natSessionEntry 13 }

**natSessionPrivateDstPort OBJECT-TYPE**

SYNTAX       InetPortNumber

MAX-ACCESS   read-only

STATUS       current

## DESCRIPTION

"When the value of protocol is TCP or UDP, this object represents the destination port in the first packet of session while in private-realm. On the other hand, when the protocol is ICMP, this object is not relevant and should be set to zero.

The value of this object must be zero when the natSessionPrivateDstEPBindId object has a zero value and natSessionPrivateDstEPBindMode is set to addressPortBind(2). In such a case, the NAT session lookup will match any port number to this field.

The value of this object must be zero when the object is not a representative field (SrcPort, DstPort, or ICMP identifier) of the session tuple in either the public realm or the private realm."

::= { natSessionEntry 14 }

## natSessionPublicAddrType OBJECT-TYPE

SYNTAX InetAddressType

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"This object specifies the address type used for natSessionPublicSrcAddr and natSessionPublicDstAddr."

::= { natSessionEntry 15 }

## natSessionPublicSrcAddr OBJECT-TYPE

SYNTAX InetAddress

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The source IP address of the session endpoint that lies in the public network.

The value of this object must be zero when the natSessionPrivateSrcEPBindId object has a zero value. In such a scenario, the NAT session lookup will match any IP address to this field.

The type of this address is determined by the value of the natSessionPublicAddrType object."

::= { natSessionEntry 16 }

## natSessionPublicSrcPort OBJECT-TYPE

SYNTAX InetPortNumber

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"When the value of protocol is TCP or UDP, this object represents the source port in the first packet of session while in public-realm. On the other hand, when protocol is ICMP, a NAT session is created only for query/response type ICMP messages such as ICMP echo, Timestamp, and Information request messages, and this object represents the public-realm specific identifier in the ICMP message, as defined in RFC 792 for ICMPv4 and in RFC 2463 for ICMPv6.

The value of this object must be zero when the natSessionPrivateSrcEPBindId object has a zero value and natSessionPrivateSrcEPBindMode is set to addressPortBind(2). In such a scenario, the NAT session lookup will match any port number to this field.

The value of this object must be zero when the object is not a representative field (SrcPort, DstPort or ICMP identifier) of the session tuple in either the public realm or the private realm."

::= { natSessionEntry 17 }

natSessionPublicDstAddr OBJECT-TYPE

SYNTAX InetAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The destination IP address of the session endpoint that lies in the public network.

The value of this object must be non-zero when the natSessionPrivateDstEPBindId object has a non-zero value. If the value of this object and the corresponding natSessionPrivateDstEPBindId object value is zero, then the NAT session lookup will match any IP address to this field.

The type of this address is determined by the value of the natSessionPublicAddrType object."

::= { natSessionEntry 18 }

natSessionPublicDstPort OBJECT-TYPE

SYNTAX InetPortNumber

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"When the value of protocol is TCP or UDP, this object represents the destination port in the first packet of session while in public-realm. On the other hand, when the protocol is ICMP, this object is not relevant for translation and should be zero.

The value of this object must be zero when the natSessionPrivateDstEPBindId object has a zero value and natSessionPrivateDstEPBindMode is addressPortBind(2). In such a scenario, the NAT session lookup will match any port number to this field.

The value of this object must be zero when the object is not a representative field (SrcPort, DstPort, or ICMP identifier) of the session tuple in either the public realm or the private realm."

::= { natSessionEntry 19 }

## natSessionMaxIdleTime OBJECT-TYPE

SYNTAX TimeTicks

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The max time for which this session can be idle without detecting a packet."

::= { natSessionEntry 20 }

## natSessionCurrentIdleTime OBJECT-TYPE

SYNTAX TimeTicks

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The time since a packet belonging to this session was last detected."

::= { natSessionEntry 21 }

## natSessionInTranslates OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The number of inbound packets that were translated for this session.

Discontinuities in the value of this counter can occur at reinitialization of the management system and at other

times, as indicated by the value of  
 ifCounterDiscontinuityTime on the relevant interface."  
 ::= { natSessionEntry 22 }

natSessionOutTranslates OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of outbound packets that were translated for  
 this session.

Discontinuities in the value of this counter can occur at  
 reinitialization of the management system and at other  
 times, as indicated by the value of  
 ifCounterDiscontinuityTime on the relevant interface."

::= { natSessionEntry 23 }

--

-- The Protocol table

--

natProtocolTable OBJECT-TYPE

SYNTAX SEQUENCE OF NatProtocolEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The (conceptual) table containing per protocol NAT  
 statistics."

::= { natMIBObjects 10 }

natProtocolEntry OBJECT-TYPE

SYNTAX NatProtocolEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry (conceptual row) containing NAT statistics  
 pertaining to a particular protocol."

INDEX { natProtocol }

::= { natProtocolTable 1 }

NatProtocolEntry ::= SEQUENCE {

natProtocol NatProtocolType,

natProtocolInTranslates Counter64,

natProtocolOutTranslates Counter64,

natProtocolDiscards Counter64

}

```
natProtocol      OBJECT-TYPE
    SYNTAX      NatProtocolType
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This object represents the protocol pertaining to which
         parameters are reported."
    ::= { natProtocolEntry 1 }

natProtocolInTranslates OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of inbound packets pertaining to the protocol
         identified by natProtocol that underwent NAT.

         Discontinuities in the value of this counter can occur at
         reinitialization of the management system and at other
         times, as indicated by the value of
         ifCounterDiscontinuityTime on the relevant interface."
    ::= { natProtocolEntry 2 }

natProtocolOutTranslates OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of outbound packets pertaining to the protocol
         identified by natProtocol that underwent NAT.

         Discontinuities in the value of this counter can occur at
         reinitialization of the management system and at other
         times, as indicated by the value of
         ifCounterDiscontinuityTime on the relevant interface."
    ::= { natProtocolEntry 3 }

natProtocolDiscards OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of packets pertaining to the protocol
         identified by natProtocol that had to be
         rejected/dropped due to lack of resources.  These
         rejections could be due to session timeout, resource
         unavailability, lack of address space, etc."
```

```
        Discontinuities in the value of this counter can occur at
        reinitialization of the management system and at other
        times, as indicated by the value of
        ifCounterDiscontinuityTime on the relevant interface."
 ::= { natProtocolEntry 4 }

--
-- Notifications section
--

natMIBNotifications OBJECT IDENTIFIER ::= { natMIB 0 }

--
-- Notifications
--

natPacketDiscard NOTIFICATION-TYPE
    OBJECTS { ifIndex }
    STATUS current
    DESCRIPTION
        "This notification is generated when IP packets are
        discarded by the NAT function; e.g., due to lack of
        mapping space when NAT is out of addresses or ports.

        Note that the generation of natPacketDiscard
        notifications is throttled by the agent, as specified
        by the 'natNotifThrottlingInterval' object."
 ::= { natMIBNotifications 1 }

--
-- Conformance information.
--

natMIBConformance OBJECT IDENTIFIER ::= { natMIB 2 }

natMIBGroups          OBJECT IDENTIFIER ::= { natMIBConformance 1 }
natMIBCompliances     OBJECT IDENTIFIER ::= { natMIBConformance 2 }

--
-- Units of conformance
--

natConfigGroup OBJECT-GROUP
    OBJECTS { natInterfaceRealm,
              natInterfaceServiceType,
              natInterfaceStorageType,
              natInterfaceRowStatus,
              natAddrMapName,
```

```

natAddrMapEntryType,
natAddrMapTranslationEntity,
natAddrMapLocalAddrType,
natAddrMapLocalAddrFrom,
natAddrMapLocalAddrTo,
natAddrMapLocalPortFrom,
natAddrMapLocalPortTo,
natAddrMapGlobalAddrType,
natAddrMapGlobalAddrFrom,
natAddrMapGlobalAddrTo,
natAddrMapGlobalPortFrom,
natAddrMapGlobalPortTo,
natAddrMapProtocol,
natAddrMapStorageType,
natAddrMapRowStatus,
natBindDefIdleTimeout,
natUdpDefIdleTimeout,
natIcmpDefIdleTimeout,
natOtherDefIdleTimeout,
natTcpDefIdleTimeout,
natTcpDefNegTimeout,
natNotifThrottlingInterval }

```

STATUS current

DESCRIPTION

"A collection of configuration-related information  
required to support management of devices supporting  
NAT."

```
 ::= { natMIBGroups 1 }
```

natTranslationGroup OBJECT-GROUP

```

OBJECTS { natAddrBindNumberOfEntries,
natAddrBindGlobalAddrType,
natAddrBindGlobalAddr,
natAddrBindId,
natAddrBindTranslationEntity,
natAddrBindType,
natAddrBindMapIndex,
natAddrBindSessions,
natAddrBindMaxIdleTime,
natAddrBindCurrentIdleTime,
natAddrBindInTranslates,
natAddrBindOutTranslates,
natAddrPortBindNumberOfEntries,
natAddrPortBindGlobalAddrType,
natAddrPortBindGlobalAddr,
natAddrPortBindGlobalPort,
natAddrPortBindId,
natAddrPortBindTranslationEntity,

```

```

natAddrPortBindType,
natAddrPortBindMapIndex,
natAddrPortBindSessions,
natAddrPortBindMaxIdleTime,
natAddrPortBindCurrentIdleTime,
natAddrPortBindInTranslates,
natAddrPortBindOutTranslates,
natSessionPrivateSrcEPBindId,
natSessionPrivateSrcEPBindMode,
natSessionPrivateDstEPBindId,
natSessionPrivateDstEPBindMode,
natSessionDirection,
natSessionUpTime,
natSessionAddrMapIndex,
natSessionProtocolType,
natSessionPrivateAddrType,
natSessionPrivateSrcAddr,
natSessionPrivateSrcPort,
natSessionPrivateDstAddr,
natSessionPrivateDstPort,
natSessionPublicAddrType,
natSessionPublicSrcAddr,
natSessionPublicSrcPort,
natSessionPublicDstAddr,
natSessionPublicDstPort,
natSessionMaxIdleTime,
natSessionCurrentIdleTime,
natSessionInTranslates,
natSessionOutTranslates }

```

STATUS current

#### DESCRIPTION

"A collection of BIND-related objects required to support management of devices supporting NAT."

::= { natMIBGroups 2 }

#### natStatsInterfaceGroup OBJECT-GROUP

```

OBJECTS { natInterfaceInTranslates,
          natInterfaceOutTranslates,
          natInterfaceDiscards }

```

STATUS current

#### DESCRIPTION

"A collection of NAT statistics associated with the interface on which NAT is configured, to aid troubleshooting/monitoring of the NAT operation."

::= { natMIBGroups 3 }

#### natStatsProtocolGroup OBJECT-GROUP

```

OBJECTS { natProtocolInTranslates,
          natProtocolOutTranslates,
          natProtocolDiscards }
STATUS   current
DESCRIPTION
    "A collection of protocol specific NAT statistics,
     to aid troubleshooting/monitoring of NAT operation."
 ::= { natMIBGroups 4 }

natStatsAddrMapGroup OBJECT-GROUP
OBJECTS { natAddrMapInTranslates,
          natAddrMapOutTranslates,
          natAddrMapDiscards,
          natAddrMapAddrUsed }
STATUS   current
DESCRIPTION
    "A collection of address map specific NAT statistics,
     to aid troubleshooting/monitoring of NAT operation."
 ::= { natMIBGroups 5 }

natMIBNotificationGroup NOTIFICATION-GROUP
NOTIFICATIONS { natPacketDiscard }
STATUS         current
DESCRIPTION
    "A collection of notifications generated by
     devices supporting this MIB."
 ::= { natMIBGroups 6 }

--
-- Compliance statements
--

natMIBFullCompliance MODULE-COMPLIANCE
STATUS   current
DESCRIPTION
    "When this MIB is implemented with support for
     read-create, then such an implementation can claim
     full compliance.  Such devices can then be both
     monitored and configured with this MIB.

     The following index objects cannot be added as OBJECT
     clauses but nevertheless have the compliance
     requirements:
     "
    -- OBJECT  natAddrBindLocalAddrType
    -- SYNTAX  InetAddressType { ipv4(1), ipv6(2) }
    -- DESCRIPTION
    --          "An implementation is required to support

```

```

--          global IPv4 and/or IPv6 addresses, depending
--          on its support for IPv4 and IPv6."

-- OBJECT  natAddrBindLocalAddr
-- SYNTAX  InetAddress (SIZE(4|16))
-- DESCRIPTION
--          "An implementation is required to support
--          global IPv4 and/or IPv6 addresses, depending
--          on its support for IPv4 and IPv6."

-- OBJECT  natAddrPortBindLocalAddrType
-- SYNTAX  InetAddressType { ipv4(1), ipv6(2) }
-- DESCRIPTION
--          "An implementation is required to support
--          global IPv4 and/or IPv6 addresses, depending
--          on its support for IPv4 and IPv6."

-- OBJECT  natAddrPortBindLocalAddr
-- SYNTAX  InetAddress (SIZE(4|16))
-- DESCRIPTION
--          "An implementation is required to support
--          global IPv4 and/or IPv6 addresses, depending
--          on its support for IPv4 and IPv6."

```

MODULE IF-MIB -- The interfaces MIB, RFC2863

```

MANDATORY-GROUPS {
    ifCounterDiscontinuityGroup
}

```

MODULE -- this module

```

MANDATORY-GROUPS { natConfigGroup, natTranslationGroup,
    natStatsInterfaceGroup }

```

```

GROUP          natStatsProtocolGroup

```

```

DESCRIPTION

```

```

    "This group is optional."

```

```

GROUP          natStatsAddrMapGroup

```

```

DESCRIPTION

```

```

    "This group is optional."

```

```

GROUP          natMIBNotificationGroup

```

```

DESCRIPTION

```

```

    "This group is optional."

```

```

OBJECT  natAddrMapLocalAddrType

```

```

SYNTAX  InetAddressType { ipv4(1), ipv6(2) }

```

```

DESCRIPTION

```

```

    "An implementation is required to support global IPv4
    and/or IPv6 addresses, depending on its support

```

for IPv4 and IPv6."

OBJECT natAddrMapLocalAddrFrom  
SYNTAX InetAddress (SIZE(4|16))  
DESCRIPTION  
"An implementation is required to support global IPv4  
and/or IPv6 addresses, depending on its support  
for IPv4 and IPv6."

OBJECT natAddrMapLocalAddrTo  
SYNTAX InetAddress (SIZE(4|16))  
DESCRIPTION  
"An implementation is required to support global IPv4  
and/or IPv6 addresses, depending on its support  
for IPv4 and IPv6."

OBJECT natAddrMapGlobalAddrType  
SYNTAX InetAddressType { ipv4(1), ipv6(2) }  
DESCRIPTION  
"An implementation is required to support global IPv4  
and/or IPv6 addresses, depending on its support  
for IPv4 and IPv6."

OBJECT natAddrMapGlobalAddrFrom  
SYNTAX InetAddress (SIZE(4|16))  
DESCRIPTION  
"An implementation is required to support global IPv4  
and/or IPv6 addresses, depending on its support  
for IPv4 and IPv6."

OBJECT natAddrMapGlobalAddrTo  
SYNTAX InetAddress (SIZE(4|16))  
DESCRIPTION  
"An implementation is required to support global IPv4  
and/or IPv6 addresses, depending on its support  
for IPv4 and IPv6."

OBJECT natAddrBindGlobalAddrType  
SYNTAX InetAddressType { ipv4(1), ipv6(2) }  
DESCRIPTION  
"An implementation is required to support global IPv4  
and/or IPv6 addresses, depending on its support  
for IPv4 and IPv6."

OBJECT natAddrBindGlobalAddr  
SYNTAX InetAddress (SIZE(4|16))  
DESCRIPTION  
"An implementation is required to support global IPv4

and/or IPv6 addresses, depending on its support for IPv4 and IPv6."

OBJECT natAddrPortBindGlobalAddrType

SYNTAX InetAddressType { ipv4(1), ipv6(2) }

DESCRIPTION

"An implementation is required to support global IPv4 and/or IPv6 addresses, depending on its support for IPv4 and IPv6."

OBJECT natAddrPortBindGlobalAddr

SYNTAX InetAddress (SIZE(4|16))

DESCRIPTION

"An implementation is required to support global IPv4 and/or IPv6 addresses, depending on its support for IPv4 and IPv6."

OBJECT natSessionPrivateAddrType

SYNTAX InetAddressType { ipv4(1), ipv6(2) }

DESCRIPTION

"An implementation is required to support global IPv4 and/or IPv6 addresses, depending on its support for IPv4 and IPv6."

OBJECT natSessionPrivateSrcAddr

SYNTAX InetAddress (SIZE(4|16))

DESCRIPTION

"An implementation is required to support global IPv4 and/or IPv6 addresses, depending on its support for IPv4 and IPv6."

OBJECT natSessionPrivateDstAddr

SYNTAX InetAddress (SIZE(4|16))

DESCRIPTION

"An implementation is required to support global IPv4 and/or IPv6 addresses, depending on its support for IPv4 and IPv6."

OBJECT natSessionPublicAddrType

SYNTAX InetAddressType { ipv4(1), ipv6(2) }

DESCRIPTION

"An implementation is required to support global IPv4 and/or IPv6 addresses, depending on its support for IPv4 and IPv6."

OBJECT natSessionPublicSrcAddr

SYNTAX InetAddress (SIZE(4|16))

## DESCRIPTION

"An implementation is required to support global IPv4 and/or IPv6 addresses, depending on its support for IPv4 and IPv6."

OBJECT natSessionPublicDstAddr

SYNTAX InetAddress (SIZE(4|16))

## DESCRIPTION

"An implementation is required to support global IPv4 and/or IPv6 addresses, depending on its support for IPv4 and IPv6."

::= { natMIBCompliances 1 }

natMIBReadOnlyCompliance MODULE-COMPLIANCE

STATUS current

## DESCRIPTION

"When this MIB is implemented without support for read-create (i.e., in read-only mode), then such an implementation can claim read-only compliance. Such a device can then be monitored but cannot be configured with this MIB.

The following index objects cannot be added as OBJECT clauses but nevertheless have the compliance requirements:

"

-- OBJECT natAddrBindLocalAddrType

-- SYNTAX InetAddressType { ipv4(1), ipv6(2) }

-- DESCRIPTION

-- "An implementation is required to support  
-- global IPv4 and/or IPv6 addresses, depending  
-- on its support for IPv4 and IPv6."

-- OBJECT natAddrBindLocalAddr

-- SYNTAX InetAddress (SIZE(4|16))

-- DESCRIPTION

-- "An implementation is required to support  
-- global IPv4 and/or IPv6 addresses, depending  
-- on its support for IPv4 and IPv6."

-- OBJECT natAddrPortBindLocalAddrType

-- SYNTAX InetAddressType { ipv4(1), ipv6(2) }

-- DESCRIPTION

-- "An implementation is required to support  
-- global IPv4 and/or IPv6 addresses, depending  
-- on its support for IPv4 and IPv6."

```

-- OBJECT  natAddrPortBindLocalAddr
-- SYNTAX  InetAddress (SIZE(4|16))
-- DESCRIPTION
--          "An implementation is required to support
--          global IPv4 and/or IPv6 addresses, depending
--          on its support for IPv4 and IPv6."

MODULE IF-MIB -- The interfaces MIB, RFC2863
  MANDATORY-GROUPS {
    ifCounterDiscontinuityGroup
  }

MODULE -- this module
  MANDATORY-GROUPS { natConfigGroup, natTranslationGroup,
    natStatsInterfaceGroup }

  GROUP      natStatsProtocolGroup
  DESCRIPTION
    "This group is optional."
  GROUP      natStatsAddrMapGroup
  DESCRIPTION
    "This group is optional."
  GROUP      natMIBNotificationGroup
  DESCRIPTION
    "This group is optional."
  OBJECT natInterfaceRowStatus
  SYNTAX RowStatus { active(1) }
  MIN-ACCESS read-only
  DESCRIPTION
    "Write access is not required, and active is the only
    status that needs to be supported."

  OBJECT natAddrMapLocalAddrType
  SYNTAX InetAddressType { ipv4(1), ipv6(2) }
  MIN-ACCESS read-only
  DESCRIPTION
    "Write access is not required. An implementation is
    required to support global IPv4 and/or IPv6 addresses,
    depending on its support for IPv4 and IPv6."

  OBJECT natAddrMapLocalAddrFrom
  SYNTAX InetAddress (SIZE(4|16))
  MIN-ACCESS read-only
  DESCRIPTION
    "Write access is not required. An implementation is
    required to support global IPv4 and/or IPv6 addresses,
    depending on its support for IPv4 and IPv6."

```

OBJECT natAddrMapLocalAddrTo  
SYNTAX InetAddress (SIZE(4|16))  
MIN-ACCESS read-only  
DESCRIPTION  
    "Write access is not required. An implementation is required to support global IPv4 and/or IPv6 addresses, depending on its support for IPv4 and IPv6."

OBJECT natAddrMapGlobalAddrType  
SYNTAX InetAddressType { ipv4(1), ipv6(2) }  
MIN-ACCESS read-only  
DESCRIPTION  
    "Write access is not required. An implementation is required to support global IPv4 and/or IPv6 addresses, depending on its support for IPv4 and IPv6."

OBJECT natAddrMapGlobalAddrFrom  
SYNTAX InetAddress (SIZE(4|16))  
MIN-ACCESS read-only  
DESCRIPTION  
    "Write access is not required. An implementation is required to support global IPv4 and/or IPv6 addresses, depending on its support for IPv4 and IPv6."

OBJECT natAddrMapGlobalAddrTo  
SYNTAX InetAddress (SIZE(4|16))  
MIN-ACCESS read-only  
DESCRIPTION  
    "Write access is not required. An implementation is required to support global IPv4 and/or IPv6 addresses, depending on its support for IPv4 and IPv6."

OBJECT natAddrMapRowStatus  
SYNTAX RowStatus { active(1) }  
MIN-ACCESS read-only  
DESCRIPTION  
    "Write access is not required, and active is the only status that needs to be supported."

OBJECT natAddrBindGlobalAddrType  
SYNTAX InetAddressType { ipv4(1), ipv6(2) }  
DESCRIPTION  
    "An implementation is required to support global IPv4 and/or IPv6 addresses, depending on its support for IPv4 and IPv6."

OBJECT natAddrBindGlobalAddr  
SYNTAX InetAddress (SIZE(4|16))

## DESCRIPTION

"An implementation is required to support global IPv4 and/or IPv6 addresses, depending on its support for IPv4 and IPv6."

OBJECT natAddrPortBindGlobalAddrType

SYNTAX InetAddressType { ipv4(1), ipv6(2) }

## DESCRIPTION

"An implementation is required to support global IPv4 and/or IPv6 addresses, depending on its support for IPv4 and IPv6."

OBJECT natAddrPortBindGlobalAddr

SYNTAX InetAddress (SIZE(4|16))

## DESCRIPTION

"An implementation is required to support global IPv4 and/or IPv6 addresses, depending on its support for IPv4 and IPv6."

OBJECT natSessionPrivateAddrType

SYNTAX InetAddressType { ipv4(1), ipv6(2) }

## DESCRIPTION

"An implementation is required to support global IPv4 and/or IPv6 addresses, depending on its support for IPv4 and IPv6."

OBJECT natSessionPrivateSrcAddr

SYNTAX InetAddress (SIZE(4|16))

## DESCRIPTION

"An implementation is required to support global IPv4 and/or IPv6 addresses, depending on its support for IPv4 and IPv6."

OBJECT natSessionPrivateDstAddr

SYNTAX InetAddress (SIZE(4|16))

## DESCRIPTION

"An implementation is required to support global IPv4 and/or IPv6 addresses, depending on its support for IPv4 and IPv6."

OBJECT natSessionPublicAddrType

SYNTAX InetAddressType { ipv4(1), ipv6(2) }

## DESCRIPTION

"An implementation is required to support global IPv4 and/or IPv6 addresses, depending on its support for IPv4 and IPv6."

OBJECT natSessionPublicSrcAddr

SYNTAX InetAddress (SIZE(4|16))

DESCRIPTION

"An implementation is required to support global IPv4 and/or IPv6 addresses, depending on its support for IPv4 and IPv6."

OBJECT natSessionPublicDstAddr

SYNTAX InetAddress (SIZE(4|16))

DESCRIPTION

"An implementation is required to support global IPv4 and/or IPv6 addresses, depending on its support for IPv4 and IPv6."

::= { natMIBCompliances 2 }

END

## 6. Acknowledgements

The authors of the document would like to thank Randy Turner, Ashwini S.T., Kevin Luehrs, Sam Sankoorikal, and Juergen Quittek for their valuable feedback.

The authors would like to especially thank Juergen Schoenwaelder for his patient and fine-combed review and detailed comments as a MIB doctor. The NAT MIB is much clearer and flatter as a result of Juergen's suggestions.

## 7. Security Considerations

It is clear that this MIB can potentially be useful for configuration. Unauthorized access to the write-able objects could cause a denial of service and/or widespread network disturbance. Hence, the support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations.

At this writing, no security holes have been identified beyond those that SNMP Security is itself intended to address. These relate primarily to controlled access to sensitive information and the ability to configure a device - or which might result from operator error, which is beyond the scope of any security architecture.

There are a number of managed objects in this MIB that may contain information that may be sensitive from a business perspective, in that they may represent NAT bind and session information. The NAT bind and session objects reveal the identity of private hosts that are engaged in a session with external end nodes. A curious outsider

could monitor these two objects to assess the number of private hosts being supported by the NAT device. Further, a disgruntled former employee of an enterprise could use the NAT bind and session information to break into specific private hosts by intercepting the existing sessions or originating new sessions into the host. There are no objects that are sensitive in their own right, such as passwords or monetary amounts. It may even be important to control GET access to these objects and possibly to encrypt the values of these objects when they are sent over the network via SNMP. Not all versions of SNMP provide features for such a secure environment.

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.

It is recommended that the implementers consider the security features as provided by the SNMPv3 framework (see [RFC3410], 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.

## 8. References

### 8.1. Normative References

- [RFC2578] McCloaghrie, K., Perkins, D., and J. Schoenwaelder, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.
- [RFC2579] McCloaghrie, K., Perkins, D., and J. Schoenwaelder, "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.
- [RFC2580] McCloaghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.
- [RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.

- [RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.
- [RFC4001] Daniele, M., Haberman, B., Routhier, S., Schoenwaelder, J., "Textual Conventions for Internet Network Addresses", RFC 4001, February 2005.
- [RFC792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, September 1981.
- [RFC3489] Rosenberg, J., Weinberger, J., Huitema, C., and R. Mahy, "STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)", RFC 3489, March 2003.
- [RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.
- [RFC2463] Conta, A. and S. Deering, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 2463, December 1998.
- [RFC3411] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", STD 62, RFC 3411, December 2002.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3413] Levi, D., Meyer, P., and B. Stewart, "Simple Network Management Protocol (SNMP) Applications", STD 62, RFC 3413, December 2002.

## 8.2. Informative References

- [RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.
- [RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

## Authors' Addresses

R. Rohit  
Mascon Global Limited  
#59/2 100 ft Ring Road  
Banashankari II Stage  
Bangalore 560 070  
India

Phone: +91 80 679 6227  
EMail: rrohit74@hotmail.com

P. Srisuresh  
Caymas Systems, Inc.  
1179-A North McDowell Blvd.  
Petaluma, CA 94954

Phone: (707) 283-5063  
EMail: srisuresh@yahoo.com

Rajiv Raghunarayan  
Cisco Systems Inc.  
170 West Tasman Drive  
San Jose, CA 95134

Phone: +1 408 853 9612  
EMail: raraghun@cisco.com

Nalinaksh Pai  
Cisco Systems, Inc.  
Prestige Waterford  
No. 9, Brunton Road  
Bangalore - 560 025  
India

Phone: +91 80 532 1300 extn. 6354  
EMail: npai@cisco.com

Cliff Wang  
Information Security  
Bank One Corp  
1111 Polaris Pkwy  
Columbus, OH 43240

Phone: +1 614 213 6117  
EMail: cliffwang2000@yahoo.com

## Full Copyright Statement

Copyright (C) The Internet Society (2005).

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](mailto:ietf-ipr@ietf.org).

## Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

