

## RIP Version 2 MIB Extension

### Status of this Memo

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

### Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in TCP/IP-based internets. In particular, it defines objects for managing RIP Version 2.

### Table of Contents

1. The Network Management Framework .....	1
2. Objects .....	2
2.1 Format of Definitions .....	2
3. Overview .....	3
3.1 Textual Conventions .....	3
3.2 Structure of MIB .....	3
4. Definitions .....	3
4.1 Global Counters .....	4
4.2 RIP Interface Tables .....	4
4.3 Peer Table .....	10
5. Acknowledgements .....	12
6. References .....	12
7. Security Considerations .....	13
8. Authors' Addresses .....	13

### 1. The Network Management Framework

The Internet-standard Network Management Framework consists of three components. They are:

STD 16/RFC 1155 which defines the SMI, the mechanisms used for describing and naming objects for the purpose of management. STD 16/RFC 1212 defines a more concise description mechanism, which is

wholly consistent with the SMI.

RFC 1156 which defines MIB-I, the core set of managed objects for the Internet suite of protocols. STD 17/RFC 1213 defines MIB-II, an evolution of MIB-I based on implementation experience and new operational requirements.

STD 15/RFC 1157 which defines the SNMP, the protocol used for network access to managed objects.

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

## 2. Objects

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the subset of Abstract Syntax Notation One (ASN.1) [7] defined in the SMI. In particular, each object has a name, a syntax, and an encoding. The name is an object identifier, an administratively assigned name, which specifies an object type. The object type together with an object instance serves to uniquely identify a specific instantiation of the object. For human convenience, we often use a textual string, termed the OBJECT DESCRIPTOR, to also refer to the object type.

The syntax of an object type defines the abstract data structure corresponding to that object type. The ASN.1 language is used for this purpose. However, the SMI [3] purposely restricts the ASN.1 constructs which may be used. These restrictions are explicitly made for simplicity.

The encoding of an object type is simply how that object type is represented using the object type's syntax. Implicitly tied to the notion of an object type's syntax and encoding is how the object type is represented when being transmitted on the network.

The SMI specifies the use of the basic encoding rules of ASN.1 [8], subject to the additional requirements imposed by the SNMP.

### 2.1 Format of Definitions

Section 4 contains contains the specification of all object types contained in this MIB module. The object types are defined using the conventions defined in the SMI, as amended by the extensions specified in [9].

### 3. Overview

#### 3.1 Textual Conventions

Several new data types are introduced as a textual convention in this MIB document. These textual conventions enhance the readability of the specification and can ease comparison with other specifications if appropriate. It should be noted that the introduction of the these textual conventions has no effect on either the syntax nor the semantics of any managed objects. The use of these is merely an artifact of the explanatory method used. Objects defined in terms of one of these methods are always encoded by means of the rules that define the primitive type. Hence, no changes to the SMI or the SNMP are necessary to accommodate these textual conventions which are adopted merely for the convenience of readers and writers in pursuit of the elusive goal of clear, concise, and unambiguous MIB documents.

The new data types are: Validation (the standard "set to invalid causes deletion" type), and RouteTag. The RouteTag type represents the contents of the Route Tag field in the packet header or route entry.

#### 3.2 Structure of MIB

The RIP-2 MIB contains global counters useful for detecting the deleterious effects of RIP incompatibilities, an "interfaces" table which contains interface-specific statistics and configuration information, and an optional "neighbor" table containing information that may be helpful in debugging neighbor relationships. Like the protocol itself, this MIB takes great care to preserve compatibility with RIP-1 systems, and controls for monitoring and controlling system interactions.

### 4. Definitions

```
RFC1389-MIB DEFINITIONS ::= BEGIN

IMPORTS
    Counter, TimeTicks, IPAddress
        FROM RFC1155-SMI
    mib-2
        FROM RFC1213-MIB
    OBJECT-TYPE
        FROM RFC-1212;

-- RIP-2 Management Information Base

rip2 OBJECT IDENTIFIER ::= { mib-2 23 }
```

```
-- the RouteTag type represents the contents of the
-- Route Tag field in the packet header or route entry.

RouteTag ::= OCTET STRING (SIZE (2))

-- the Validation type is used for the variable that deletes
-- an entry from a table, and ALWAYS takes at least these values:

Validation ::= INTEGER { valid (1), invalid (2) }

--      The RIP-2 Globals Group.
--      Implementation of this group is mandatory for systems that
--      implement RIP-2.

-- These counters are intended to facilitate debugging quickly
-- changing routes or failing neighbors

rip2GlobalGroup OBJECT IDENTIFIER ::= { rip2 1 }

rip2GlobalRouteChanges OBJECT-TYPE
    SYNTAX      Counter
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "The number of changes made to the IP Route Da-
        tabase by RIP."
    ::= { rip2GlobalGroup 1 }

rip2GlobalQueries OBJECT-TYPE
    SYNTAX      Counter
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "The number of responses sent to RIP queries
        from other systems."
    ::= { rip2GlobalGroup 2 }

-- RIP Interfaces Groups
-- Implementation of these Groups is mandatory for systems that
-- implement RIP-2.

-- Since RIP versions 1 and 2 do not deal with addressless links,
-- it is assumed that RIP "interfaces" are subnets within a
-- routing domain.
```

-- The RIP Interface Status Table.

```
rip2IfStatTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Rip2IfStatEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION
        "A list of subnets which require separate
        status monitoring in RIP."
    ::= { rip2 2 }
```

```
rip2IfStatEntry OBJECT-TYPE
    SYNTAX      Rip2IfStatEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION
        "A Single Routing Domain in a single Subnet."
    INDEX { rip2IfStatAddress }
    ::= { rip2IfStatTable 1 }
```

```
Rip2IfStatEntry ::=
    SEQUENCE {
        rip2IfStatAddress
            IpAddress,
        rip2IfStatRcvBadPackets
            Counter,
        rip2IfStatRcvBadRoutes
            Counter,
        rip2IfStatSentUpdates
            Counter,
        rip2IfStatStatus
            Validation
    }
```

```
rip2IfStatAddress OBJECT-TYPE
    SYNTAX      IpAddress
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "The IP Address of this system on the indicated
        subnet."
    ::= { rip2IfStatEntry 1 }
```

```
rip2IfStatRcvBadPackets OBJECT-TYPE
    SYNTAX      Counter
    ACCESS      read-only
```

STATUS mandatory  
DESCRIPTION  
"The number of RIP response packets received by  
the RIP process which were subsequently dis-  
carded for any reason (e.g. a version 0 packet,  
or an unknown command type)."  
::= { rip2IfStatEntry 2 }

rip2IfStatRcvBadRoutes OBJECT-TYPE  
SYNTAX Counter  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The number of routes, in valid RIP packets,  
which were ignored for any reason (e.g. unknown  
address family, or invalid metric)."  
::= { rip2IfStatEntry 3 }

rip2IfStatSentUpdates OBJECT-TYPE  
SYNTAX Counter  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The number of triggered RIP updates actually  
sent on this interface. This explicitly does  
NOT include full updates sent containing new  
information."  
::= { rip2IfStatEntry 4 }

rip2IfStatStatus OBJECT-TYPE  
SYNTAX Validation  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION  
"Writing invalid has the effect of deleting  
this interface."  
DEFVAL { valid }  
::= { rip2IfStatEntry 5 }

-- The RIP Interface Configuration Table.

rip2IfConfTable OBJECT-TYPE  
SYNTAX SEQUENCE OF Rip2IfConfEntry  
ACCESS not-accessible

```

    STATUS    mandatory
    DESCRIPTION
        "A list of subnets which require separate con-
        figuration in RIP."
    ::= { rip2 3 }

rip2IfConfEntry OBJECT-TYPE
    SYNTAX    Rip2IfConfEntry
    ACCESS    not-accessible
    STATUS    mandatory
    DESCRIPTION
        "A Single Routing Domain in a single Subnet."
    INDEX { rip2IfConfAddress }
    ::= { rip2IfConfTable 1 }

Rip2IfConfEntry ::=
    SEQUENCE {
        rip2IfConfAddress
            IpAddress,
        rip2IfConfDomain
            RouteTag,
        rip2IfConfAuthType
            INTEGER,
        rip2IfConfAuthKey
            OCTET STRING (SIZE(0..16)),
        rip2IfConfSend
            INTEGER,
        rip2IfConfReceive
            INTEGER,
        rip2IfConfDefaultMetric
            INTEGER,
        rip2IfConfStatus
            Validation
    }

rip2IfConfAddress OBJECT-TYPE
    SYNTAX    IpAddress
    ACCESS    read-only
    STATUS    mandatory
    DESCRIPTION
        "The IP Address of this system on the indicated
        subnet."
    ::= { rip2IfConfEntry 1 }

rip2IfConfDomain OBJECT-TYPE
    SYNTAX    RouteTag

```

```

ACCESS    read-write
STATUS    mandatory
DESCRIPTION
    "Value inserted into the Routing Domain field
    of all RIP packets sent on this interface."
DEFVAL { '0000'h }
::= { rip2IfConfEntry 2 }

```

```

rip2IfConfAuthType OBJECT-TYPE
    SYNTAX      INTEGER {
                    noAuthentication (1),
                    simplePassword (2)
                }
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "The type of Authentication used on this inter-
        face."
    DEFVAL { noAuthentication }
    ::= { rip2IfConfEntry 3 }

```

```

rip2IfConfAuthKey OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(0..16))
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "The value to be used as the Authentication Key
        whenever the corresponding instance of
        rip2IfConfAuthType has the value simplePass-
        word. A modification of the corresponding in-
        stance of rip2IfConfAuthType does not modify
        the rip2IfConfAuthKey value.

```

If a string shorter than 16 octets is supplied, it will be left-justified and padded to 16 octets, on the right, with nulls (0x00).

Reading this object always results in an OCTET STRING of length zero; authentication may not be bypassed by reading the MIB object."

```

DEFVAL { ''h }
::= { rip2IfConfEntry 4 }

```

```

rip2IfConfSend OBJECT-TYPE
    SYNTAX      INTEGER {

```



```

        doNotSend (1),
        ripVersion1 (2),
        rip1Compatible (3),
        ripVersion2 (4)
    }
ACCESS    read-write
STATUS    mandatory
DESCRIPTION
    "What the router sends on this interface.
    ripVersion1 implies sending RIP updates compli-
    ant with RFC 1058.  rip1Compatible implies
    broadcasting RIP-2 updates using RFC 1058 route
    subsumption rules.  ripVersion2 implies multi-
    casting RIP-2 updates."
DEFVAL { rip1Compatible }
 ::= { rip2IfConfEntry 5 }

```

```

rip2IfConfReceive OBJECT-TYPE
    SYNTAX    INTEGER {
                rip1 (1),
                rip2 (2),
                rip1OrRip2 (3)
            }
    ACCESS    read-write
    STATUS    mandatory
    DESCRIPTION
        "This indicates which version of RIP updates
        are to be accepted.  Note that rip2 and
        rip1OrRip2 implies reception of multicast pack-
        ets."
    DEFVAL { rip1OrRip2 }
    ::= { rip2IfConfEntry 6 }

```

```

rip2IfConfDefaultMetric OBJECT-TYPE
    SYNTAX    INTEGER ( 0..15 )
    ACCESS    read-write
    STATUS    mandatory
    DESCRIPTION
        "This variable indicates what metric is to be
        used as a default route in RIP updates ori-
        ginated on this interface.  A value of zero in-
        dicates that no default route should be ori-
        ginated; in this case, a default route via
        another router may be propagated."
    ::= { rip2IfConfEntry 7 }

```

```
rip2IfConfStatus OBJECT-TYPE
    SYNTAX      Validation
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Writing invalid has the effect of deleting
         this interface."
    DEFVAL { valid }
    ::= { rip2IfConfEntry 8 }
```

-- Peer Table

-- The RIP Peer Group  
-- Implementation of this Group is Optional

-- This group provides information about active peer  
-- relationships intended to assist in debugging.

```
rip2PeerTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Rip2PeerEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION
        "A list of RIP Peers."
    ::= { rip2 4 }
```

```
rip2PeerEntry OBJECT-TYPE
    SYNTAX      Rip2PeerEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION
        "Information regarding a single routing peer."
    INDEX { rip2PeerAddress, rip2PeerDomain }
    ::= { rip2PeerTable 1 }
```

```
Rip2PeerEntry ::=
    SEQUENCE {
        rip2PeerAddress
            IpAddress,
        rip2PeerDomain
            RouteTag,
        rip2PeerLastUpdate
            TimeTicks,
        rip2PeerVersion
            INTEGER,
        rip2PeerRcvBadPackets
```

```
        Counter,  
    rip2PeerRcvBadRoutes  
        Counter  
    }
```

```
rip2PeerAddress OBJECT-TYPE  
    SYNTAX      IpAddress  
    ACCESS      read-only  
    STATUS      mandatory  
    DESCRIPTION  
        "The IP Address of the Peer System."  
    ::= { rip2PeerEntry 1 }
```

```
rip2PeerDomain OBJECT-TYPE  
    SYNTAX      RouteTag  
    ACCESS      read-only  
    STATUS      mandatory  
    DESCRIPTION  
        "The value in the Routing Domain field in RIP  
        packets received from the peer."  
    ::= { rip2PeerEntry 2 }
```

```
rip2PeerLastUpdate OBJECT-TYPE  
    SYNTAX      TimeTicks  
    ACCESS      read-only  
    STATUS      mandatory  
    DESCRIPTION  
        "The value of sysUpTime when the most recent  
        RIP update was received from this system."  
    ::= { rip2PeerEntry 3 }
```

```
rip2PeerVersion OBJECT-TYPE  
    SYNTAX      INTEGER ( 0..255 )  
    ACCESS      read-only  
    STATUS      mandatory  
    DESCRIPTION  
        "The RIP version number in the header of the  
        last RIP packet received."  
    ::= { rip2PeerEntry 4 }
```

```
rip2PeerRcvBadPackets OBJECT-TYPE  
    SYNTAX      Counter  
    ACCESS      read-only
```

```
STATUS    mandatory
DESCRIPTION
    "The number of RIP response packets from this
    peer discarded as invalid."
 ::= { rip2PeerEntry 5 }
```

```
rip2PeerRcvBadRoutes OBJECT-TYPE
    SYNTAX      Counter
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "The number of routes from this peer that were
        ignored because the entry format was invalid."
    ::= { rip2PeerEntry 6 }
```

END

## 5. Acknowledgements

This document was produced by the RIP-2 Working Group of the Internet Engineering Task Force (IETF).

In addition, the comments of the following individuals are also acknowledged: Keith McCloghrie and Frank Kastenholz.

## 8. References

- [1] Cerf, V., "IAB Recommendations for the Development of Internet Network Management Standards", RFC 1052, IAB, April 1988.
- [2] Cerf, V., "Report of the Second Ad Hoc Network Management Review Group", RFC 1109, IAB, August 1989.
- [3] Rose M., and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based internets", STD 16, RFC 1155, Performance Systems International, Hughes LAN Systems, May 1990.
- [4] McCloghrie K., and M. Rose, "Management Information Base for Network Management of TCP/IP-based internets", RFC 1156, Hughes LAN Systems, Performance Systems International, May 1990.
- [5] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol", STD 15, RFC 1157, SNMP Research, Performance Systems International, Performance Systems International, MIT Laboratory for Computer Science, May 1990.

- [6] Rose, M., Editor, "Management Information Base for Network Management of TCP/IP-based internets: MIB-II", RFC 1158, Performance Systems International, May 1990.
- [7] Information processing systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1), International Organization for Standardization, International Standard 8824, December 1987.
- [8] Information processing systems - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Notation One (ASN.1), International Organization for Standardization, International Standard 8825, December 1987.
- [9] Rose, M., and K. McCloghrie, Editors, "Concise MIB Definitions", STD 16, RFC 1212, Performance Systems International, Hughes LAN Systems, March 1991.
- [10] Malkin, G., "RIP Version 2 - Carrying Additional Information", RFC 1388, Xylogics, Inc., January 1993.
- [11] Malkin, G., "RIP Version 2 Protocol Analysis", RFC 1387, Xylogics, Inc., January 1993.

## 7. Security Considerations

Security issues are not discussed in this memo.

## 8. Authors' Addresses

Gary Malkin  
Xylogics, Inc.  
53 Third Avenue  
Burlington, MA 01803

Phone: (617) 272-8140  
EMail: gmalkin@Xylogics.COM

Fred Baker  
Advanced Computer Communications  
315 Bollay Drive  
Santa Barbara, California 93117-6014

Phone: (805) 685-4455  
EMail: fbaker@acc.com