

**Party MIB
for Version 2 of the
Simple Network Management Protocol (SNMPv2)**

19 March 1995

[draft-ietf-snmpv2-party-ds-01.txt](#)

Jeffrey D. Case
SNMP Research, Inc.
case@snmp.com

James Galvin
Trusted Information Systems
galvin@tis.com

Keith McCloghrie
Cisco Systems, Inc.
kzm@cisco.com

Marshall T. Rose
Dover Beach Consulting, Inc.
mrose@dbc.mtview.ca.us

Steven Waldbusser
Carnegie Mellon University
waldbusser@cmu.edu

Status of this Memo

This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as ``work in progress.''

To learn the current status of any Internet-Draft, please check the ``1id-abstracts.txt'' listing contained in the Internet-Drafts Shadow

Expires September 1995

[Page 1]

Directories on ds.internic.net (US East Coast), nic.nordu.net (Europe),
ftp.isi.edu (US West Coast), or munnari.oz.au (Pacific Rim).

1. Introduction

A management system contains: several (potentially many) nodes, each with a processing entity, termed an agent, which has access to management instrumentation; at least one management station; and, a management protocol, used to convey management information between the agents and management stations. Operations of the protocol are carried out under an administrative framework which defines authentication, authorization, access control, and privacy policies.

Management stations execute management applications which monitor and control managed elements. Managed elements are devices such as hosts, routers, terminal servers, etc., which are monitored and controlled via access to their management information.

Management information is viewed as a collection of managed objects, residing in a virtual information store, termed the Management Information Base (MIB). Collections of related objects are defined in MIB modules. These modules are written using a subset of OSI's Abstract Syntax Notation One (ASN.1) [[1](#)], termed the Structure of Management Information (SMI) [[2](#)].

The Administrative Infrastructure for SNMPv2 document [[3](#)] defines the properties associated with SNMPv2 parties, SNMPv2 contexts, and access control policies. It is the purpose of this document, the Party MIB for SNMPv2, to define managed objects which correspond to these properties.

1.1. A Note on Terminology

For the purpose of exposition, the original Internet-standard Network Management Framework, as described in RFCs 1155, 1157, and 1212, is termed the SNMP version 1 framework (SNMPv1). The current framework is termed the SNMP version 2 framework (SNMPv2).

1.2. Change Log

For the 19 March version: +

- The many changes adopted by the SNMPv2 Working Group. +

For the 1 November version:

Expires September 1995

[Page 3]

- recast [RFC 1447](#) into an Internet-Draft,
- fixed typos,
- added snmpUDPDomain to the IMPORTS clause,
- added descriptive text for the initial Context conventions,
- added text to the DESCRIPTIONs of partyStatus, contextStatus, acStatus, and viewStatus, specifying that columnar objects can be modified without having to set the status column to 'notInService',
- added clarifying text to the DESCRIPTIONs of contextLocal, contextLocalEntity, contextProxySrcParty, contextProxyDstParty, contextProxyContext, and contextStatus,
- removed auxiliary objects from the definition of partyMIBGroup.

2. Definitions

SNMPv2-PARTY-MIB DEFINITIONS ::= BEGIN

IMPORTS

MODULE-IDENTITY, OBJECT-IDENTITY, OBJECT-TYPE, snmpModules,
UInteger32, zeroDotZero
FROM SNMPv2-SMI
snmpUDPDomain
FROM SNMPv2-TM
TEXTUAL-CONVENTION, RowStatus, TruthValue
FROM SNMPv2-TC
MODULE-COMPLIANCE, OBJECT-GROUP
FROM SNMPv2-CONF;

partyMIB MODULE-IDENTITY

LAST-UPDATED "9503190000Z"
ORGANIZATION "IETF SNMPv2 Working Group"
CONTACT-INFO
" Keith McCloghrie

Postal: Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA

Phone: +1 408 526 5260
Email: kzm@cisco.com"

DESCRIPTION

"The MIB module describing SNMPv2 parties."

REVISION "9104300000Z"

DESCRIPTION

"The initial revision of this MIB module was published as
[RFC 1447](#)."

::= { snmpModules 3 }

Expires September 1995

[Page 5]

-- textual conventions

Party ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"Denotes a SNMPv2 party identifier.

Note that agents may impose implementation limitations on the length of OIDs used to identify Parties. As such, management stations creating new parties should be aware that using an excessively long OID may result in the agent refusing to perform the set operation and instead returning the appropriate error response, e.g., noCreation."

SYNTAX OBJECT IDENTIFIER

TAddress ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"Denotes a transport service address.

For snmpUDPDomain, a TAddress is 6 octets long, the initial 4 octets containing the IP-address in network-byte order and the last 2 containing the UDP port in network-byte order. Consult [5] for further information on snmpUDPDomain."

SYNTAX OCTET STRING

Clock ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"A party's authentication clock - a non-negative integer which is incremented as specified/allowed by the party's Authentication Protocol.

For noAuth, a party's authentication clock is unused and its value is undefined.

For v2md5AuthProtocol, a party's authentication clock is a relative clock with 1-second granularity."

SYNTAX UInteger32 (0..2147483647)

|

Expires September 1995

[Page 6]

Context ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"Denotes a SNMPv2 context identifier.

Note that agents may impose implementation limitations on the length of OIDs used to identify Contexts. As such, management stations creating new contexts should be aware that using an excessively long OID may result in the agent refusing to perform the set operation and instead returning the appropriate error response, e.g., noCreation."

SYNTAX OBJECT IDENTIFIER

AccessPrivileges ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"A set of access privileges which specify the authorized set of management operations between two SNMPv2 entities.

These privileges are specified as a sum of values, where each value specifies a SNMPv2 PDU type by which the subject party may request a permitted operation. The value for a particular PDU type is computed as 2 raised to the value of the ASN.1 context-specific tag for the appropriate SNMPv2 PDU type. The values (for the tags defined in [5]) are defined in [3] as:

| | | |
|-------------|---|-----|
| Get | : | 1 |
| GetNext | : | 2 |
| Response | : | 4 |
| Set | : | 8 |
| unused | : | 16 |
| GetBulk | : | 32 |
| Inform | : | 64 |
| SNMPv2-Trap | : | 128 |

The null set is represented by the value zero."

SYNTAX INTEGER (0..255)

Expires September 1995

[Page 7]

StorageType ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"Describes the memory realization of a conceptual row. A row which is volatile(2) is lost upon reboot. A row which is either nonVolatile(3), permanent(4) or readOnly(5), is backed up by stable storage. A row which is permanent(4) can be changed but not deleted. A row which is readOnly(5) cannot be changed nor deleted.

If the value of an object with this syntax is either permanent(4) or readOnly(5), it cannot be modified. Conversely, if the value is either other(1), volatile(2) or nonVolatile(3), it cannot be modified to be permanent(4) or readOnly(5).

Every usage of this textual convention is required to specify the columnar objects which a permanent(4) row must at a minimum allow to be writable."

SYNTAX

```
INTEGER {
    other(1),      -- eh?
    volatile(2),   -- e.g., in RAM
    nonVolatile(3), -- e.g., in NVRAM
    permanent(4),  -- e.g., partially in ROM
    readOnly(5)    -- e.g., completely in ROM
}
```

Expires September 1995

[Page 8]

-- administrative assignments

partyAdmin OBJECT IDENTIFIER ::= { partyMIB 1 }

-- definitions of security protocols

partyProtocols OBJECT IDENTIFIER ::= { partyAdmin 1 }

noAuth OBJECT-IDENTITY
 STATUS current
 DESCRIPTION
 "The protocol without authentication."
 ::= { partyProtocols 1 }

noPriv OBJECT-IDENTITY
 STATUS current
 DESCRIPTION
 "The protocol without privacy."
 ::= { partyProtocols 2 }

-- defined in [\[4\]](#)
desPrivProtocol OBJECT-IDENTITY
 STATUS current
 DESCRIPTION
 "The DES Privacy Protocol."
 ::= { partyProtocols 3 }

-- defined in [\[4\]](#)
v2md5AuthProtocol OBJECT-IDENTITY
 STATUS current
 DESCRIPTION
 "The MD5 Authentication Protocol."
 ::= { partyProtocols 4 }

Expires September 1995

[Page 9]

-- definitions of temporal domains

temporalDomains

OBJECT IDENTIFIER ::= { partyAdmin 2 }

currentTime OBJECT-IDENTITY

STATUS current

DESCRIPTION

"The temporal domain which refers to management information
at the current time."

::= { temporalDomains 1 }

restartTime OBJECT-IDENTITY

STATUS current

DESCRIPTION

"The temporal domain which refers to management information
upon the next re-initialization of the managed device."

::= { temporalDomains 2 }

cacheTime OBJECT-IDENTITY

STATUS current

DESCRIPTION

"The prefix for temporal domains which refer to management
information that is cached. In particular, the temporal
domain:

{ cacheTime N }

and guaranteed to be at most N seconds old."

::= { temporalDomains 3 }

Expires September 1995

[Page 10]

```
-- object assignments -

partyMIBObjects
    OBJECT IDENTIFIER ::= { partyMIB 2 }

-- SNMPv2 party information |

snmpParties    OBJECT IDENTIFIER ::= { partyMIBObjects 1 }

partyTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF PartyEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "The SNMPv2 Party database."
    ::= { snmpParties 1 }

partyEntry OBJECT-TYPE
    SYNTAX      PartyEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "Locally held information about a particular SNMPv2 party."
    INDEX       { IMPLIED partyIdentity }
    ::= { partyTable 1 }
```

Expires September 1995

[Page 11]

PartyEntry ::=

```

SEQUENCE {
    partyIdentity      Party,
    partyIndex         INTEGER,
    partyTDomain       OBJECT IDENTIFIER,
    partyTAddress      TAddress,
    partyMaxMessageSize INTEGER,
    partyLocal         TruthValue,
    partyAuthProtocol  OBJECT IDENTIFIER,
    partyAuthClock     Clock,
    partyAuthPrivate   OCTET STRING,
    partyAuthPublic    OCTET STRING,
    partyAuthLifetime  INTEGER,
    partyPrivProtocol  OBJECT IDENTIFIER,
    partyPrivPrivate   OCTET STRING,
    partyPrivPublic    OCTET STRING,
    partyCloneFrom     Party,
    partyStorageType   StorageType,
    partyStatus        RowStatus,
    partyAuthChange    OCTET STRING,
    partyPrivChange    OCTET STRING
}

```

partyIdentity OBJECT-TYPE

SYNTAX Party

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A party identifier uniquely identifying a particular SNMPv2
 party. This object is prohibited from taking the value { 0
 0 }."

::= { partyEntry 1 }

Expires September 1995

[Page 12]

partyIndex OBJECT-TYPE

SYNTAX INTEGER (1..2147483647) |

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"An arbitrary unique value for each SNMPv2 party. |

The value for each SNMPv2 party must remain constant at
least from one re-initialization of the entity's network
management system to the next re-initialization. |The specific value is meaningful only within a given SNMPv2
entity, i.e., it is not meaningful to any other SNMPv2
entity except to uniquely identify the party within the set
of all parties known to this agent." |

::= { partyEntry 2 }

partyTDomain OBJECT-TYPE

SYNTAX OBJECT IDENTIFIER

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Indicates the kind of transport service by which the party
receives network management traffic."

DEFVAL { snmpUDPDomain }

::= { partyEntry 3 }

partyTAddress OBJECT-TYPE

SYNTAX TAddress

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The transport service address by which the party receives
network management traffic, formatted according to the
corresponding value of partyTDomain. The agent must listen +
for SNMPv2 requests sent to this address. The default +
assignment of a transport-layer 'port' is that specified by +
the standard Transport Mapping for the kind of transport +
service given by the corresponding value of partyTDomain. +For snmpUDPDomain, partyTAddress is formatted as a 4-octet
IP Address concatenated with a 2-octet UDP port number."

DEFVAL { '000000000000'H }

::= { partyEntry 4 }

Expires September 1995

[Page 13]

partyMaxMessageSize OBJECT-TYPE

SYNTAX INTEGER (484..65507)

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The maximum length in octets of a SNMPv2 message which this party will accept. For parties which execute at an agent, the agent initializes this object to the maximum length supported by the agent, and does not let the object be set to any larger value. For parties which do not execute at the agent, the agent must allow the manager to set this object to any legal value, even if it is larger than the agent can generate."

DEFVAL { 484 }

::= { partyEntry 5 }

partyLocal OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"An indication of whether this party executes at this SNMPv2 entity. If this object has a value of true(1), then the SNMPv2 entity will listen for SNMPv2 messages on the partyTAddress associated with this party. If this object has the value false(2), then the SNMPv2 entity will not listen for SNMPv2 messages on the partyTAddress associated with this party."

DEFVAL { false }

::= { partyEntry 6 }

Expires September 1995

[Page 14]

partyAuthProtocol OBJECT-TYPE

SYNTAX OBJECT IDENTIFIER

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The authentication protocol by which all messages generated by the party are authenticated as to origin and integrity. The value noAuth signifies that messages generated by the party are not authenticated.

An instance of this object is created concurrently with the creation of any other object instance for the same party (i.e., as part of the processing of the set operation which creates the first object instance in the same conceptual row). Once created, the value of an instance of this object can not be changed."

DEFVAL { v2md5AuthProtocol }

::= { partyEntry 7 }

partyAuthClock OBJECT-TYPE

SYNTAX Clock

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The authentication clock which represents the local notion of the current time specific to the party. This value must not be decremented unless the party's private authentication key is changed simultaneously."

DEFVAL { 0 }

::= { partyEntry 8 }

Expires September 1995

[Page 15]

partyAuthPrivate OBJECT-TYPE

SYNTAX OCTET STRING

-- for v2md5AuthProtocol: (SIZE (16))

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"An encoding of the party's private authentication key which may be needed to support the authentication protocol. Although the value of this variable may be altered by a management operation (e.g., a SNMPv2 Set-Request), its value can never be retrieved by a management operation: when read, the value of this variable is the zero length OCTET STRING.

The private authentication key is NOT directly represented by the value of this variable, but rather it is represented according to an encoding. This encoding is the bitwise exclusive-OR of the old key with the new key, i.e., of the old private authentication key (prior to the alteration) with the new private authentication key (after the alteration). Thus, when processing a received protocol Set operation, the new private authentication key is obtained from the value of this variable as the result of a bitwise exclusive-OR of the variable's value and the old private authentication key. In calculating the exclusive-OR, if the old key is shorter than the new key, zero-valued padding is appended to the old key. If no value for the old key exists, a zero-length OCTET STRING is used in the calculation."

DEFVAL { 'H } -- the empty string

::= { partyEntry 9 }

Expires September 1995

[Page 16]

partyAuthPublic OBJECT-TYPE

SYNTAX OCTET STRING

-- for v2md5AuthProtocol: (SIZE (0..16))

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"A publically-readable value for the party.

Depending on the party's authentication protocol, this value may be needed to support the party's authentication protocol. Alternatively, it may be used by a manager during the procedure for altering secret information about a party. (For example, by altering the value of an instance of this object in the same SNMPv2 Set-Request used to update an instance of partyAuthPrivate, a subsequent Get-Request can determine if the Set-Request was successful in the event that no response to the Set-Request is received, see [4].)

The length of the value is dependent on the party's authentication protocol. If not used by the authentication protocol, it is recommended that agents support values of any length up to and including the length of the corresponding partyAuthPrivate object."

DEFVAL { 'H } -- the empty string

::= { partyEntry 10 }

partyAuthLifetime OBJECT-TYPE

SYNTAX INTEGER (0..2147483647)

UNITS "seconds"

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The lifetime (in units of seconds) which represents an administrative upper bound on acceptable delivery delay for protocol messages generated by the party.

An instance of this object is created concurrently with the creation of any other object instance for the same party (i.e., as part of the processing of the set operation which creates the first object instance in the same conceptual row). Once created, the value of an instance of this object can not be changed."

DEFVAL { 300 }

::= { partyEntry 11 }

Expires September 1995

[Page 17]

partyPrivProtocol OBJECT-TYPE

SYNTAX OBJECT IDENTIFIER

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The privacy protocol by which all protocol messages received by the party are protected from disclosure. The value noPriv signifies that messages received by the party are not protected.

An instance of this object is created concurrently with the creation of any other object instance for the same party (i.e., as part of the processing of the set operation which creates the first object instance in the same conceptual row). Once created, the value of an instance of this object can not be changed."

DEFVAL { noPriv }

::= { partyEntry 12 }

partyPrivPrivate OBJECT-TYPE

SYNTAX OCTET STRING

-- for desPrivProtocol: (SIZE (16))

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"An encoding of the party's private encryption key which may be needed to support the privacy protocol. Although the value of this variable may be altered by a management operation (e.g., a SNMPv2 Set-Request), its value can never be retrieved by a management operation: when read, the value of this variable is the zero length OCTET STRING.

The private encryption key is NOT directly represented by the value of this variable, but rather it is represented according to an encoding. This encoding is the bitwise exclusive-OR of the old key with the new key, i.e., of the old private encryption key (prior to the alteration) with the new private encryption key (after the alteration). Thus, when processing a received protocol Set operation, the new private encryption key is obtained from the value of this variable as the result of a bitwise exclusive-OR of the variable's value and the old private encryption key. In calculating the exclusive-OR, if the old key is shorter than the new key, zero-valued padding is appended to the old key. If no value for the old key exists, a zero-length OCTET STRING is used in the calculation."

DEFVAL { 'H } -- the empty string

::= { partyEntry 13 }

Expires September 1995

[Page 19]

partyPrivPublic OBJECT-TYPE

SYNTAX OCTET STRING

-- for desPrivProtocol: (SIZE (0..16))

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"A publically-readable value for the party.

Depending on the party's privacy protocol, this value may be needed to support the party's privacy protocol.

Alternatively, it may be used by a manager as a part of its procedure for altering secret information about a party.

(For example, by altering the value of an instance of this object in the same SNMPv2 Set-Request used to update an instance of partyPrivPrivate, a subsequent Get-Request can determine if the Set-Request was successful in the event that no response to the Set-Request is received, see [\[4\]](#).)

The length of the value is dependent on the party's privacy protocol. If not used by the privacy protocol, it is recommended that agents support values of any length up to and including the length of the corresponding partyPrivPrivate object."

DEFVAL { 'H } -- the empty string

::= { partyEntry 14 }

Expires September 1995

[Page 20]

partyCloneFrom OBJECT-TYPE

SYNTAX Party
MAX-ACCESS read-create
STATUS current
DESCRIPTION

"The identity of a party to clone authentication and privacy parameters from. When read, the value { 0 0 } is returned.

This value must be written exactly once, when the associated instance of partyStatus either does not exist or has the value 'notReady'. When written, the value identifies a party, the cloning party, whose status column has the value 'active'. The cloning party is used in two ways.

One, if instances of the following objects do not exist for the party being created, then they are created with values identical to those of the corresponding objects for the cloning party:

partyAuthProtocol
partyAuthPublic
partyAuthLifetime
partyPrivProtocol
partyPrivPublic

Two, instances of the following objects are updated using the corresponding values of the cloning party:

partyAuthPrivate
partyPrivPrivate

(e.g., the value of the cloning party's instance of the partyAuthPrivate object is XOR'd with the value of the partyAuthPrivate instances of the party being created.)"

::= { partyEntry 15 }

Expires September 1995

[Page 21]

partyStorageType OBJECT-TYPE

SYNTAX StorageType

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The storage type for this conceptual row in the partyTable. |

Conceptual rows having the value 'permanent' must allow |
write-access at a minimum to: partyTDomain and partyTAddress |
for all parties; partyAuthClock, partyAuthPrivate and |
partyAuthPublic for parties employing authentication; and |
partyPrivPrivate and partyPrivPublic for parties employing |
privacy. |

Note that any party which employs authentication or privacy |
must allow its clock and secrets to be updated and thus |
cannot be 'readOnly'." |

DEFVAL { nonVolatile }

::= { partyEntry 16 }

partyStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The status of this conceptual row in the partyTable. |

A party is not qualified for activation until instances of |
all columns of its partyEntry row have an appropriate value. |
In particular, one or more management set operations are |
required to configure the party: |

a value must be written to the party's partyCloneFrom |
object, and |

if the value of the party's partyAuthProtocol object is |
not noAuth, |
then the corresponding instance of partyAuthPrivate must |
contain a secret of the appropriate length. Further, at |
least one management protocol set operation updating the |
value of the party's partyAuthPrivate object must be |
successfully processed before the partyAuthPrivate column |
is considered appropriately configured, and |

if the value of party's partyPrivProtocol object is not |

Expires September 1995

[Page 22]

```

noPriv,
then the corresponding instance of partyPrivPrivate must
contain a secret of the appropriate length. Further, at
least one management protocol set operation updating the
value of the party's partyPrivPrivate object must be
successfully processed
before the partyPrivPrivate column is considered
appropriately configured.

```

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

```

For those columnar objects which permit write-access (after
their initial creation),
their value in an existing conceptual row can be changed
irrespective of the value of partyStatus for that row."

```

```

::= { partyEntry 17 }

```

```

partyAuthChange  OBJECT-TYPE
    SYNTAX      OCTET STRING      -- typically (SIZE (0..32))
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "A manager-generated, partially-random value which, when
        modified, causes the corresponding instance of
        partyAuthPrivate to be modified via a one-way function.

        The value of an instance of this object is the concatenation
        of two components: a 'random' component and a 'delta'
        component. The lengths of the random and delta components
        are given by the corresponding value of partyAuthProtocol;
        for authentication protocols requiring partyAuthPrivate to
        be a fixed length, the length of both the random and delta
        components is that fixed length; for authentication
        protocols allowing the length of partyAuthPrivate to be
        variable up to a particular maximum length, the length of
        the random component is that maximum length and the length
        of the delta component is any length less than or equal to
        the maximum length. For example, v2md5AuthProtocol requires
        a fixed length of 16 octets, and for the purposes of this
        definition, noAuth and rfc1157noAuth allow a variable length
        up to a maximum of 16 octets. Other authentication
        protocols may define other sizes, as deemed appropriate.

```

Expires September 1995

[Page 23]

When an instance of this object is modified to have a new value by the management protocol, the agent generates a new value of the corresponding instance of partyAuthPrivate as follows:

- the existing value of the corresponding instance of partyAuthPrivate is extended if necessary with zero-value padding to be the same length as the random component;
- the new value of the random component is appended to the (possibly extended) value of the corresponding instance of partyAuthPrivate, and the result is subjected to the MD5 hash algorithm to produce a digest value;
- this digest value, truncated if necessary to be the same length as the new value of the delta component, is XOR-ed with the new value of the delta component to produce the new value of the corresponding instance of partyAuthPrivate.

i.e.,

```
keyIntermediate = md5(keyold XOR randomComponent)
keynew = deltaComponent XOR keyIntermediate
```

The value of this object whenever it is retrieved by the management protocol is always the zero-length string."

```
DEFVAL { 'H }      -- the empty string
::= { partyEntry 18 }
```

partyPrivChange OBJECT-TYPE

SYNTAX OCTET STRING -- typically (SIZE (0..32))

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"A manager-generated, partially-random value which, when modified, causes the corresponding instance of partyPrivPrivate to be modified via a one-way function.

The value of an instance of this object is the concatenation of two components: a 'random' component and a 'delta' component. The lengths of the random and delta components are given by the corresponding value of partyPrivProtocol; for privacy protocols requiring partyPrivPrivate to be a fixed length, the length of both the random and delta components is that fixed length; for privacy protocols allowing the length of partyPrivPrivate to be variable up to

Expires September 1995

[Page 24]

a particular maximum length, the length of the random component is that maximum length and the length of the delta component is any length less than or equal to the maximum length. For example, desPrivProtocol requires a fixed length of 16 octets, and for the purposes of this definition, noPriv allows a variable length up to a maximum of 16 octets. Other privacy protocols may define other sizes, as deemed appropriate.

When an instance of this object is modified to have a new value by the management protocol, the agent generates a new value of the corresponding instance of partyPrivPrivate as follows:

- the existing value of the corresponding instance of partyPrivPrivate is extended if necessary with zero-value padding to be the same length as the random component;
- the new value of the random component is appended to the (possibly extended) value of the corresponding instance of partyPrivPrivate, and the result is subjected to the MD5 hash algorithm to produce a digest value;
- this digest value, truncated if necessary to be the same length as the new value of the delta component, is XOR-ed with the new value of the delta component to produce the new value of the corresponding instance of partyPrivPrivate.

i.e.,

```
keyIntermediate = md5(keyold XOR randomComponent)
keynew = deltaComponent XOR keyIntermediate
```

The value of this object whenever it is retrieved by the management protocol is always the zero-length string."

```
DEFVAL { 'H }      -- the empty string
 ::= { partyEntry 19 }
```

```
partySecretSpinLock  OBJECT-TYPE
    SYNTAX      TestAndIncr
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "An advisory lock used to allow several cooperating SNMPv2
        entities, all acting in a manager role, to coordinate their
```

Expires September 1995

[Page 25]

```
        use of facilities to alter secrets in the Party Table."      +
 ::= { snmpParties 2 }                                              +
```



```

-- Agent Identifier |
agentID OBJECT-TYPE +
    SYNTAX          OCTET STRING (SIZE (12)) +
    MAX-ACCESS      read-write +
    STATUS          current +
    DESCRIPTION      +
        "The agent's administratively-unique identifier. +

        The initial value for this object may be configured via an +
        operator console entry or via an algorithmic function. In +
        the later case, the following guidelines are recommended: +

        1) The first four octets should be set to the binary +
           equivalent of the agent's SNMP network management +
           private enterprise number as assigned by the Internet +
           Assigned Numbers Authority (IANA). For example, if +
           Acme Networks has been assigned { enterprises 696 }, +
           the first four octets would be assigned '000002b8'H. +

        2) The remaining eight octets are the cookie whose +
           contents are determined via one or more enterprise- +
           specific methods. Such methods must be designed so as +
           to maximize the possibility that the value of this +
           object will be unique in the agent's administrative +
           domain. For example, the cookie may be the IP address +
           of the agent, or the MAC address of one of the +
           interfaces, with each address suitably padded with +
           random octets. If multiple methods are defined, then +
           it is recommended that the cookie be further divided +
           into one octet that indicates the method being used and +
           seven octets which are a function of the method. +

        If set by a management operation, the value must be +
        persistent across re-initializations of the entity's network +
        management system." +
 ::= { snmpParties 3 } +

```

Expires September 1995

[Page 27]

```
-- SNMPv2 contexts
```

```
+
```

```
snmpContexts    OBJECT IDENTIFIER ::= { partyMIBObjects 2 }
```

```
contextTable OBJECT-TYPE
```

```
    SYNTAX      SEQUENCE OF ContextEntry
```

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

```
    STATUS      current
```

```
    DESCRIPTION
```

```
        "The SNMPv2 Context database."
```

```
    ::= { snmpContexts 1 }
```

```
contextEntry OBJECT-TYPE
```

```
    SYNTAX      ContextEntry
```

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

```
    STATUS      current
```

```
    DESCRIPTION
```

```
        "Locally held information about a particular SNMPv2  
        context."
```

```
    INDEX       { IMPLIED contextIdentity }
```

```
    ::= { contextTable 1 }
```

```
ContextEntry ::=
```

```
    SEQUENCE {
```

| | | |
|----------------------|--------------------|---|
| contextIdentity | Context, | |
| contextIndex | INTEGER, | |
| contextLocalEntity | OCTET STRING, | - |
| contextLocalTime | OBJECT IDENTIFIER, | |
| contextProxyDstParty | Party, | |
| contextProxySrcParty | Party, | |
| contextProxyContext | OBJECT IDENTIFIER, | |
| contextStorageType | StorageType, | |
| contextStatus | RowStatus, | |
| contextType | INTEGER | |

```
}
```

Expires September 1995

[Page 28]

contextIdentity OBJECT-TYPE

SYNTAX Context

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A context identifier uniquely identifying a particular
SNMPv2 context. This object is prohibited from taking the
value { 0 x } for any value of x."

::= { contextEntry 1 }

contextIndex OBJECT-TYPE

SYNTAX INTEGER (1..2147483647)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"An arbitrary unique value for each SNMPv2 context.
The value for each SNMPv2 context must remain constant at
least from one re-initialization of the entity's network
management system to the next re-initialization."

The specific value is meaningful only within a given SNMPv2
entity, i.e., it is not meaningful to any other SNMPv2
entity except to uniquely identify the context within the
set of all contexts known to this agent."

::= { contextEntry 2 }

-- ::= { contextEntry 3 } this OID is obsolete

-- ::= { contextEntry 4 } this OID is obsolete

Expires September 1995

[Page 29]

contextLocalEntity OBJECT-TYPE

SYNTAX OCTET STRING

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"If the value of the corresponding instance of the contextType is local(1), then the value of an instance of this object uniquely identifies the local entity (e.g., a logical device managed by the same agent) whose management information is in the SNMPv2 context's MIB view. The empty string indicates that the MIB view contains the SNMPv2 entity's own local management information; otherwise, a non-empty string indicates that the MIB view contains management information of some other local entity, e.g., 'Repeater1'.

If the value of the corresponding instance of the contextType is remote(2), then the value of an instance of this object identifies an entity which is local to the SNMPv2 entity which realizes this SNMPv2 context, and whose management information is in the SNMPv2 context's MIB view.

If the value of the corresponding instance of the contextType is proxy(3), then the value of an instance of this object is ignored and can take any value at the agent's discretion, e.g., the zero-length string."

DEFVAL { 'H } -- the empty string

::= { contextEntry 5 }

contextLocalTime OBJECT-TYPE

SYNTAX OBJECT IDENTIFIER

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"If the value of the corresponding instance of the contextType is local(1) or remote(2), then the value of an instance of this object identifies the temporal context of the management information in the MIB view.

If the value of the corresponding instance of the contextType is proxy(3), then the value of an instance of this object is ignored and can take any value at the agent's discretion, e.g., { 0 0 }."

DEFVAL { currentTime }

Expires September 1995

[Page 30]


```
::= { contextEntry 6 }
```

contextProxyDstParty OBJECT-TYPE

SYNTAX Party

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"If the corresponding instance of contextType has the value proxy(3), and if requests referencing this context are to be forwarded, then the value of an instance of this object identifies the proxy destination party to be used in such forwarded requests. Otherwise, the value of an instance of this object is ignored and can take any value at the agent's discretion, e.g., { 0 0 }."

DEFVAL { zeroDotZero }

```
::= { contextEntry 7 }
```

contextProxySrcParty OBJECT-TYPE

SYNTAX Party

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"If the corresponding instance of contextType has the value proxy(3), and if requests referencing this context are to be forwarded, then the value of an instance of this object identifies the proxy source party to be used in such forwarded requests. Otherwise, the value of an instance of this object is ignored and can take any value at the agent's discretion, e.g., { 0 0 }."

Interpretation of an instance of this object depends upon the transport domain of the proxy destination party (i.e., upon the value of partyTDomain for the party identified by the corresponding instance of contextProxyDstParty)."

DEFVAL { zeroDotZero }

```
::= { contextEntry 8 }
```

Expires September 1995

[Page 31]

contextProxyContext OBJECT-TYPE

SYNTAX OBJECT IDENTIFIER

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"If the corresponding instance of contextType has the value proxy(3), and if requests referencing this context are to be forwarded, then the value of an instance of this object identifies the context to be used in such forwarded requests. Otherwise, the value of an instance of this object is ignored and can take any value at the agent's discretion, e.g., { 0 0 }.

Interpretation of an instance of this object depends upon the value of the transport domain associated with the SNMPv2 party used as the proxy destination for this proxy context."

DEFVAL { zeroDotZero }

::= { contextEntry 9 }

contextStorageType OBJECT-TYPE

SYNTAX StorageType

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The storage type for this conceptual row in the contextTable. Conceptual rows having the value 'permanent' need not allow write-access to any columnar objects in the row, unless they are proxy contexts in which case they must allow at a minimum write-access to: contextProxyDstParty, contextProxySrcParty and contextProxyContext."

DEFVAL { nonVolatile }

::= { contextEntry 10 }

Expires September 1995

[Page 32]

contextStatus OBJECT-TYPE

SYNTAX RowStatus
 MAX-ACCESS read-create
 STATUS current
 DESCRIPTION

"The status of this conceptual row in the contextTable.

A context is not qualified for activation until instances of |
 all corresponding columns have consistent values. |

For those columnar objects which permit write-access, their -
 value in an existing conceptual row can be changed
 irrespective of the value of contextStatus for that row."

::= { contextEntry 11 }

contextType OBJECT-TYPE

SYNTAX INTEGER { local(1), remote(2), proxy(3) }
 MAX-ACCESS read-create
 STATUS current
 DESCRIPTION

"The type of context.

local(1) - this conceptual row refers to a SNMPv2 context +
 containing MIB views of a locally accessible entity; +
 the value of the corresponding instances of the +
 contextLocalEntity and contextLocalTime objects provide +
 further information on the local entity and its +
 temporal domain. +

remote(2) - this conceptual row refers to a SNMPv2 context +
 which is realized by a remote SNMPv2 entity. +

proxy(3) - this conceptual row refers to a SNMPv2 proxy +
 context; the values of the corresponding instances of +
 the contextProxyDstParty, contextProxySrcParty, and +
 contextProxyContext objects provide further information +
 on the proxied context and the parties used to access +
 it." +

DEFVAL { local }

::= { contextEntry 12 }

Expires September 1995

[Page 33]

```
-- SNMPv2 access privileges |
snmpAccess    OBJECT IDENTIFIER ::= { partyMIBObjects 3 }
-- ::= { snmpAccess 1 }      this OID is obsolete +
acTable OBJECT-TYPE |
    SYNTAX      SEQUENCE OF AcEntry |
    MAX-ACCESS   not-accessible |
    STATUS      current |
    DESCRIPTION |
        "The access privileges database." |
    ::= { snmpAccess 2 } |
acEntry OBJECT-TYPE |
    SYNTAX      AcEntry |
    MAX-ACCESS   not-accessible |
    STATUS      current |
    DESCRIPTION |
        "Each conceptual row in this table represents an individual |
        access policy, called an ACL (for historical reasons). |
        |
        An ACL specifies the access privileges authorized for |
        communication from one SNMPv2 party to another concerning |
        information contained in a particular SNMPv2 context. |
        |
        For each conceptual row in this table which is retained |
        across a re-initialization of the entity's network |
        management system, the combination of the partyIdentity |
        values of the referenced parties and the contextIdentity |
        value of the referenced context must be the same after the |
        re-initialization as it was before the re-initialization, |
        even if the values of acTarget, acSubject and acContext |
        vary." |
    INDEX      { acTarget, acSubject, acContext } |
    ::= { acTable 1 } |
```

Expires September 1995

[Page 34]


```
AcEntry ::=
    SEQUENCE {
        acTarget          INTEGER,
        acSubject         INTEGER,
        acContext         INTEGER,
        acPrivileges      AccessPrivileges,
        acReadViewIndex   INTEGER,
        acWriteViewIndex  INTEGER,
        acStorageType     StorageType,
        acStatus          RowStatus
    }

acTarget          OBJECT-TYPE
    SYNTAX         INTEGER (1..2147483647)
    MAX-ACCESS     not-accessible
    STATUS         current
    DESCRIPTION
        "The value of partyIndex for one of the SNMPv2 parties
        between which communication is authorized concerning
        information contained in a particular SNMPv2 context.  In
        particular, for retrieval or set operations, this object
        identifies the destination party; for notification
        operations, this object identifies the source party."
    ::= { acEntry 1 }

acSubject         OBJECT-TYPE
    SYNTAX         INTEGER (1..2147483647)
    MAX-ACCESS     not-accessible
    STATUS         current
    DESCRIPTION
        "The value of partyIndex for one of the SNMPv2 parties
        between which communication is authorized concerning
        information contained in a particular SNMPv2 context.  In
        particular, for retrieval or set operations, this object
        identifies the source party; for notification operations,
        this object identifies the destination party."
    ::= { acEntry 2 }
```

Expires September 1995

[Page 35]

```
acContext OBJECT-TYPE                                     |
    SYNTAX      INTEGER (1..2147483647)                  |
    MAX-ACCESS   not-accessible                           |
    STATUS       current                                   |
    DESCRIPTION  |
        "The value of this instance identifies the SNMPv2 context |
        associated with a particular set of access privileges, |
        and has the same value as the instance of the contextIndex |
        object for that SNMPv2 context."                   |
    ::= { acEntry 3 }                                     |
```

```

acPrivileges OBJECT-TYPE
    SYNTAX      AccessPrivileges
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "The access privileges authorized for communication between
        a particular local SNMPv2 party and a particular remote
        SNMPv2 party concerning information contained in a
        particular SNMPv2 context."
    DEFVAL       { 35 }      -- Get, Get-Next & Get-Bulk
    ::= { acEntry 4 }

acReadViewIndex OBJECT-TYPE
    SYNTAX      INTEGER (0..2147483647)
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "If, for the SNMPv2 context identified by the corresponding
        instance of acContext, the value of contextType is local(1),
        then the value of an instance of this object identifies the
        MIB view of the SNMPv2 context to which this conceptual row
        authorizes read access.  The identified MIB view is that for
        which viewIndex has the same value as the instance of this
        object; if the value is zero or there are no active view
        subtrees for that value, then the identified MIB view is the
        empty set of view subtrees.  (Note that read access includes
        access via retrieval requests as well as transmission of
        information via notification requests.)

        Otherwise, this object is ignored and can take any value at
        the agent's discretion, e.g., zero."
    DEFVAL       { 0 }
    ::= { acEntry 5 }

```

Expires September 1995

[Page 37]

acWriteViewIndex OBJECT-TYPE

SYNTAX INTEGER (0..2147483647)

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"If, for the SNMPv2 context identified by the corresponding instance of acContext, the value of contextType is local(1), then the value of an instance of this object identifies the MIB view of the SNMPv2 context to which this conceptual row authorizes write access. The identified MIB view is that for which viewIndex has the same value as the instance of this object; if the value is zero or there are no active view subtrees for that value, then the identified MIB view is the empty set of view subtrees.

Otherwise, this object is ignored and can take any value at the agent's discretion, e.g., zero."

DEFVAL { 0 }

::= { acEntry 6 }

acStorageType OBJECT-TYPE

SYNTAX StorageType

MAX-ACCESS read-create

STATUS current

DESCRIPTION

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

DEFVAL { nonVolatile }

::= { acEntry 7 }

acStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The status of this conceptual row in the acTable.

For those columnar objects which permit write-access, their value in an existing conceptual row can be changed irrespective of the value of acStatus for that row.

A conceptual row in this table is not qualified for activation until the context and both parties it references are active. Further, a conceptual row in this table is

Expires September 1995

[Page 38]

```
        immediately made notInService whenever the status of the      |
        context or either party it references is made notInService,    |
        Finally, a conceptual row in this table is immediately         |
        destroyed whenever the context or either party it references    |
        is destroyed."                                                 |
 ::= { acEntry 8 }                                                    |
```



```
-- MIB views |

snmpViews      OBJECT IDENTIFIER ::= { partyMIBObjects 4 }
```

viewNextIndex OBJECT-TYPE +

SYNTAX INTEGER (0..2147483647) +

MAX-ACCESS read-only +

STATUS current +

DESCRIPTION +

"The next unassigned value of viewIndex. The value 0 +

indicates that no unassigned values are available. +

Reading a non-zero value causes the assignment of the +

retrieved value for use as the viewIndex of a future MIB +

view, and thus causes the value of this object to change. +

The algorithm for changing nextViewIndex is implementation- +

dependent, and the agent may use a subset of values within +

1..2147483647, but the agent must guarantee that the value +

held by this object is not assigned to any in-use value of +

viewIndex, e.g., is not pointed to by any other MIB object. +

A management station creates a new MIB view using this +

algorithm: first, issue a management protocol retrieval +

operation to obtain the value of viewNextIndex - this value +

is used as the viewIndex of the new MIB view; and, second, +

issue a management protocol set operation to create an +

instance of the viewStatus object setting its value to +

`createAndGo' or `createAndWait' (as specified in the +

description of the RowStatus textual convention)." +

```
::= { snmpViews 2 } +
```

viewTable OBJECT-TYPE

SYNTAX SEQUENCE OF ViewEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Locally held information about the subtrees of MIB views |

known to this SNMPv2 entity. Note that a MIB view which has |

no subtrees defined for it has no entries in this table. |

Each SNMPv2 context which is locally accessible has zero or |

more MIB views. Each MIB view is defined by two collections |

of view subtrees: the included view subtrees, and the |

excluded view subtrees. |

Expires September 1995

[Page 40]

Every such subtree, both included and excluded, is defined in this table.

To determine if a particular object instance is in a particular MIB view, compare the object instance's OBJECT IDENTIFIER with each of the MIB view's active entries in this table. If none match, then the object instance is not in the MIB view. If one or more match, then the object instance is included in, or excluded from, the MIB view according to the value of viewType in the entry whose value of viewSubtree has the most sub-identifiers. If multiple entries match and have the same number of sub-identifiers, then the lexicographically greatest instance of viewType determines the inclusion or exclusion.

An object instance's OBJECT IDENTIFIER X matches an active entry in this table

when the number of sub-identifiers in X is at least as many as in the value of viewSubtree for the entry, and each sub-identifier in the value of viewSubtree matches its corresponding sub-identifier in X. Two sub-identifiers match either if the corresponding bit of viewMask is zero (the 'wild card' value), or if they are equal.

Due to this 'wild card' capability, we introduce the term, a 'family' of view subtrees, to refer to the set of subtrees defined by a particular combination of values of viewSubtree and viewMask. In the case where no 'wild card' is defined in viewMask, the family of view subtrees reduces to a single view subtree."

::= { snmpViews 1 }

Expires September 1995

[Page 41]

viewEntry OBJECT-TYPE

SYNTAX ViewEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION

"Information on a particular family of view subtrees included in or excluded from a particular SNMPv2 context's MIB view.

Implementations must not restrict the number of families of view subtrees for a given MIB view, except as dictated by resource constraints on the overall number of entries in the viewTable."

INDEX { viewIndex, IMPLIED viewSubtree }
 ::= { viewTable 1 }

ViewEntry ::=

SEQUENCE {
 viewIndex INTEGER,
 viewSubtree OBJECT IDENTIFIER,
 viewMask OCTET STRING,
 viewType INTEGER,
 viewStorageType StorageType,
 viewStatus RowStatus
 }

viewIndex OBJECT-TYPE

SYNTAX INTEGER (1..2147483647)
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION

"An arbitrary unique value for each MIB view.
The value for each MIB view must remain constant at least from one re-initialization of the entity's network management system to the next re-initialization.

The specific value is meaningful only within a given SNMPv2 entity, i.e., it is not meaningful to any other SNMPv2 entity except to uniquely identify the view within the set of all views known to this agent."

::= { viewEntry 1 }

Expires September 1995

[Page 42]

viewSubtree OBJECT-TYPE

SYNTAX OBJECT IDENTIFIER

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A MIB subtree."

::= { viewEntry 2 }

viewMask OBJECT-TYPE

SYNTAX OCTET STRING (SIZE (0..16))

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The bit mask which, in combination with the corresponding instance of viewSubtree, defines a family of view subtrees.

Each bit of this bit mask corresponds to a sub-identifier of viewSubtree, with the most significant bit of the i-th octet of this octet string value (extended if necessary, see below) corresponding to the $(8*i - 7)$ -th sub-identifier, and the least significant bit of the i-th octet of this octet string corresponding to the $(8*i)$ -th sub-identifier, where i is in the range 1 through 16.

Each bit of this bit mask specifies whether or not the corresponding sub-identifiers must match when determining if an OBJECT IDENTIFIER is in this family of view subtrees; a '1' indicates that an exact match must occur; a '0' indicates 'wild card', i.e., any sub-identifier value matches.

Thus, the OBJECT IDENTIFIER X of an object instance is contained in a family of view subtrees if the following criteria are met:

for each sub-identifier of the value of viewSubtree,
either:

the i-th bit of viewMask is 0, or

the i-th sub-identifier of X is equal to the i-th
sub-identifier of the value of viewSubtree.

If the value of this bit mask is M bits long and there are more than M sub-identifiers in the corresponding instance of

viewSubtree, then the bit mask is extended with 1's to be the required length.

Note that when the value of this object is the zero-length string, this extension rule results in a mask of all-1's being used (i.e., no 'wild card'), and the family of view subtrees is the one view subtree uniquely identified by the corresponding instance of viewSubtree."

```
DEFVAL      { 'H' }
::= { viewEntry 3 }
```

viewType OBJECT-TYPE

```
SYNTAX      INTEGER {
                included(1),
                excluded(2)
            }
```

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The status of a particular family of view subtrees within the particular SNMPv2 context's MIB view. The value 'included(1)' indicates that the corresponding instances of viewSubtree and viewMask define a family of view subtrees included in the MIB view. The value 'excluded(2)' indicates that the corresponding instances of viewSubtree and viewMask define a family of view subtrees excluded from the MIB view."

```
DEFVAL      { included }
::= { viewEntry 4 }
```

viewStorageType OBJECT-TYPE

```
SYNTAX      StorageType
```

MAX-ACCESS read-create

STATUS current

DESCRIPTION

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

```
DEFVAL      { nonVolatile }
::= { viewEntry 5 }
```

Expires September 1995

[Page 44]

viewStatus OBJECT-TYPE

SYNTAX RowStatus
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"The status of this conceptual row in the viewTable.

For those columnar objects which permit write-access, their
value in an existing conceptual row can be changed
irrespective of the value of viewStatus for that row."

::= { viewEntry 6 }

```
-- conformance information +

partyMIBConformance
    OBJECT IDENTIFIER ::= { partyMIB 3 }

partyMIBCompliances
    OBJECT IDENTIFIER ::= { partyMIBConformance 1 }
partyMIBGroups
    OBJECT IDENTIFIER ::= { partyMIBConformance 2 }

-- compliance statements

unSecurableCompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
        "The compliance statement for SNMPv2 entities which
        implement the Party MIB, but do not support any
        authentication or privacy protocols (i.e., only the noAuth
        and noPriv protocols are supported)."
```

```
MODULE -- this module
    MANDATORY-GROUPS { partyMIBGroup }
    ::= { partyMIBCompliances 1 }
```

```
partyNoPrivacyCompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
        "The compliance statement for SNMPv2 entities which
        implement the Party MIB, and support an authentication
        protocol, but do not support any privacy protocols (i.e.,
        only the noAuth, v2md5AuthProtocol, and noPriv protocols are
        supported)."
```

```
MODULE -- this module
    MANDATORY-GROUPS { partyMIBGroup }
    ::= { partyMIBCompliances 2 }
```

Expires September 1995

[Page 46]

partyPrivacyCompliance MODULE-COMPLIANCE

STATUS current

DESCRIPTION

"The compliance statement for SNMPv2 entities which implement the Party MIB, support an authentication protocol, and support a privacy protocol ONLY for the purpose of accessing security parameters.

For all acTable entries authorizing a local and/or remote SNMPv2 party

whose privacy protocol is desPrivProtocol, to be used in accessing a SNMPv2 context, the MIB view for that SNMPv2 context shall include only those objects subordinate to partyMIBObjects, or a subset thereof, e.g.,

```
viewSubtree = { partyMIBObjects }
viewMask    = ''H
viewType     = { included }
```

Any attempt to configure an entry in the partyTable, the contextTable, the acTable or the viewTable such that a party using the desPrivProtocol would be authorized for use in accessing objects outside of the partyMIBObjects subtree shall result in the appropriate error response (e.g., wrongValue or inconsistentValue)."

MODULE -- this module

MANDATORY-GROUPS { partyMIBGroup }

::= { partyMIBCompliances 3 }

fullPrivacyCompliance MODULE-COMPLIANCE

STATUS current

DESCRIPTION

"The compliance statement for SNMPv2 entities which implement the Party MIB, support an authentication protocol, and support a privacy protocol without restrictions on its use."

MODULE -- this module

MANDATORY-GROUPS { partyMIBGroup }

::= { partyMIBCompliances 4 }

Expires September 1995

[Page 47]

-- units of conformance

partyMIBGroup OBJECT-GROUP

```

  OBJECTS { partyIndex, partyTDomain, partyTAddress,
             partyMaxMessageSize, partyLocal,
             partyAuthProtocol, partyAuthClock,
             partyAuthPrivate, partyAuthPublic,
             partyAuthLifetime, partyPrivProtocol,
             partyPrivPrivate, partyPrivPublic,
             partyStorageType, partyStatus,
             partyCloneFrom,
             partyAuthChange, partyPrivChange,
             partySecretSpinLock,
             agentID,
             contextIndex, contextType, contextLocalEntity,
             contextLocalTime, contextStorageType,
             contextStatus,
             acPrivileges, acStorageType, acStatus,
             acReadViewIndex, acWriteViewIndex,
             viewNextIndex,
             viewMask, viewType, viewStorageType, viewStatus }

```

STATUS current

DESCRIPTION

"The collection of objects allowing the description and configuration of SNMPv2 parties.

Note that objects which support proxy contexts are not included in this conformance group."

::= { partyMIBGroups 1 }

partyMIBProxyGroup OBJECT-GROUP

```

  OBJECTS { contextProxySrcParty, contextProxyDstParty,
             contextProxyContext }

```

STATUS current

DESCRIPTION

"The collection of objects needed for the support of proxy SNMPv2 contexts."

::= { partyMIBGroups 2 }

END

Expires September 1995

[Page 48]

3. Acknowledgments

The authors wish to acknowledge the contributions of the SNMPv2 Working Group in general. In particular, the following individuals

Dave Arneson (Cabletron),
Uri Blumenthal (IBM),
Doug Book (Chipcom),
Maria Greene (Ascom Timeplex),
Deirdre Kostik (Bellcore),
Dave Harrington (Cabletron),
Jeff Johnson (Cisco Systems),
Brian O'Keefe (Hewlett Packard),
Dave Perkins (Bay Networks),
Randy Presuhn (Peer Networks),
Shawn Routhier (Epilogue),
Bob Stewart (Cisco Systems),
Kaj Tesink (Bellcore).

deserve special thanks for their contributions.

4. References

- [1] Information processing systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1), International Organization for Standardization. International Standard 8824, (December, 1987).
- [2] Case, J., McCloghrie, K., Rose, M., and Waldbusser, S., "Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)", Internet Draft, SNMP Research, Inc., Cisco Systems, Dover Beach Consulting, Inc., Carnegie Mellon University, November 1994.
- [3] Case, J., Galvin, J., McCloghrie, K., Rose, M., and Waldbusser, S., "Administrative Infrastructure for Version 2 of the Simple Network Management Protocol (SNMPv2)", Internet Draft, SNMP Research, Inc., Trusted Information Systems, Cisco Systems, Dover Beach Consulting, Inc., Carnegie Mellon University, November 1994.
- [4] Case, J., Galvin, J., McCloghrie, K., Rose, M., and Waldbusser, S., "Security Protocols for Version 2 of the Simple Network Management

Expires September 1995

[Page 49]

Protocol (SNMPv2)", Internet Draft, SNMP Research, Inc., Trusted
Information Systems, Cisco Systems, Dover Beach Consulting, Inc.,
Carnegie Mellon University,
November 1994.

- [5] Case, J., McCloghrie, K., Rose, M., and Waldbusser, S., "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", Internet Draft, SNMP Research, Inc., Cisco Systems, Dover Beach Consulting, Inc., Carnegie Mellon University, November 1994.
- [5] Case, J., McCloghrie, K., Rose, M., and Waldbusser, S., "Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)", Internet Draft, SNMP Research, Inc., Cisco Systems, Dover Beach Consulting, Inc., Carnegie Mellon University, November 1994.

5. Security Considerations

Security issues are not discussed in this memo.

6. Authors' Addresses

Jeffrey D. Case
SNMP Research, Inc.
3001 Kimberlin Heights Rd.
Knoxville, TN 37920-9716
US

Phone: +1 615 573 1434
Email: case@snmp.com

James M. Galvin
Trusted Information Systems, Inc.
3060 Washington Road, Route 97
Glenwood, MD 21738

Phone: +1 301 854-6889
EMail: galvin@tis.com

Keith McCloghrie
Cisco Systems, Inc.
170 West Tasman Drive,
San Jose CA 95134-1706.

Phone: +1 408 526 5260
Email: kzm@cisco.com

Marshall T. Rose
Dover Beach Consulting, Inc.
420 Whisman Court
Mountain View, CA 94043-2186
US

Phone: +1 415 968 1052
Email: mrose@dbc.mtview.ca.us

Expires September 1995

[Page 51]

| | |
|----------------------------|---|
| Steven Waldbusser | + |
| Carnegie Mellon University | + |
| 5000 Forbes Ave | + |
| Pittsburgh, PA 15213 | + |
| US | + |
| Phone: +1 412 268 6628 | + |
| Email: waldbusser@cmu.edu | + |

Table of Contents

| | | |
|-----------------------|----------------------------------|--------------------|
| 1 | Introduction | 3 |
| 1.1 | A Note on Terminology | 3 |
| 1.2 | Change Log | 3 |
| 2 | Definitions | 5 |
| 3.1 | Textual Conventions | 6 |
| 3.2 | Administrative Assignments | 9 |
| 3.3 | Object Assignments | 11 |
| 3.4 | SNMPv2 Party Information | 11 |
| 3.5 | Agent Identifier | 27 |
| 3.6 | SNMPv2 Contexts | 28 |
| 3.7 | SNMPv2 Access Privileges | 34 |
| 3.8 | MIB Views | 40 |
| 3.9 | Conformance Information | 46 |
| 3.9.1 | Compliance Statements | 46 |
| 3.9.2 | Units of Conformance | 48 |
| 3 | Acknowledgments | 49 |
| 4 | References | 49 |
| 5 | Security Considerations | 51 |
| 6 | Authors' Addresses | 51 |

Expires September 1995

[Page 53]