



SNMPv2-SMI

Version: 2821

Copyright 2007-2010 ImageStream Internet Solutions, Inc., All rights Reserved.

Table of Contents

SNMP/SNMPv2-SML.....	1
----------------------	---

SNMP/SNMPv2-SMI

```
SNMPv2-SMI DEFINITIONS ::= BEGIN

-- the path to the root

org          OBJECT IDENTIFIER ::= { iso 3 } -- "iso" = 1
dod          OBJECT IDENTIFIER ::= { org 6 }
internet    OBJECT IDENTIFIER ::= { dod 1 }

directory    OBJECT IDENTIFIER ::= { internet 1 }

mgmt         OBJECT IDENTIFIER ::= { internet 2 }
mib-2       OBJECT IDENTIFIER ::= { mgmt 1 }
transmission OBJECT IDENTIFIER ::= { mib-2 10 }

experimental OBJECT IDENTIFIER ::= { internet 3 }

private     OBJECT IDENTIFIER ::= { internet 4 }
enterprises OBJECT IDENTIFIER ::= { private 1 }

security     OBJECT IDENTIFIER ::= { internet 5 }

snmpV2      OBJECT IDENTIFIER ::= { internet 6 }

-- transport domains
snmpDomains OBJECT IDENTIFIER ::= { snmpV2 1 }

-- transport proxies
snmpProxys  OBJECT IDENTIFIER ::= { snmpV2 2 }

-- module identities
snmpModules OBJECT IDENTIFIER ::= { snmpV2 3 }

-- Extended UTCTime, to allow dates with four-digit years
-- (Note that this definition of ExtUTCTime is not to be IMPORTed
-- by MIB modules.)
ExtUTCTime ::= OCTET STRING(SIZE(11 | 13))
  -- format is YYMMDDHHMMZ or YYYYMMDDHHMMZ

  -- where: YY - last two digits of year (only years
  --          between 1900-1999)
  --          YYYY - last four digits of the year (any year)
  --          MM - month (01 through 12)
  --          DD - day of month (01 through 31)
  --          HH - hours (00 through 23)
  --          MM - minutes (00 through 59)
  --          Z - denotes GMT (the ASCII character Z)
  --
  -- For example, "9502192015Z" and "199502192015Z" represent
  -- 8:15pm GMT on 19 February 1995. Years after 1999 must use
  -- the four digit year format. Years 1900-1999 may use the
  -- two or four digit format.

-- definitions for information modules
```

```

MODULE-IDENTITY MACRO ::=
BEGIN
    TYPE NOTATION ::=
        "LAST-UPDATED" value(Update ExtUTCTime)
        "ORGANIZATION" Text
        "CONTACT-INFO" Text
        "DESCRIPTION" Text
        RevisionPart

    VALUE NOTATION ::=
        value(VALUE OBJECT IDENTIFIER)

    RevisionPart ::=
        Revisions
        | empty
    Revisions ::=
        Revision
        | Revisions Revision
    Revision ::=
        "REVISION" value(Update ExtUTCTime)
        "DESCRIPTION" Text

    -- a character string as defined in section 3.1.1
    Text ::= value(IA5String)
END

OBJECT-IDENTITY MACRO ::=
BEGIN
    TYPE NOTATION ::=
        "STATUS" Status
        "DESCRIPTION" Text

        ReferPart

    VALUE NOTATION ::=
        value(VALUE OBJECT IDENTIFIER)

    Status ::=
        "current"
        | "deprecated"
        | "obsolete"

    ReferPart ::=
        "REFERENCE" Text
        | empty

    -- a character string as defined in section 3.1.1
    Text ::= value(IA5String)
END

-- names of objects
-- (Note that these definitions of ObjectName and NotificationName
-- are not to be IMPORTed by MIB modules.)

ObjectName ::=
    OBJECT IDENTIFIER

NotificationName ::=
    OBJECT IDENTIFIER

```

```
-- syntax of objects

-- the "base types" defined here are:
-- 3 built-in ASN.1 types: INTEGER, OCTET STRING, OBJECT IDENTIFIER
-- 8 application-defined types: Integer32, IPAddress, Counter32,
--    Gauge32, Unsigned32, TimeTicks, Opaque, and Counter64

ObjectSyntax ::=
    CHOICE {
        simple
            SimpleSyntax,
            -- note that SEQUENCES for conceptual tables and
            -- rows are not mentioned here...

        application-wide
            ApplicationSyntax
    }

-- built-in ASN.1 types

SimpleSyntax ::=
    CHOICE {
        -- INTEGERS with a more restrictive range
        -- may also be used
        integer-value          -- includes Integer32
            INTEGER (-2147483648..2147483647),
        -- OCTET STRINGS with a more restrictive size
        -- may also be used
        string-value
            OCTET STRING (SIZE (0..65535)),
        objectID-value
            OBJECT IDENTIFIER
    }

-- indistinguishable from INTEGER, but never needs more than
-- 32-bits for a two's complement representation
Integer32 ::=
    INTEGER (-2147483648..2147483647)

-- application-wide types

ApplicationSyntax ::=
    CHOICE {
        ipAddress-value
            IPAddress,
        counter-value
            Counter32,
        timeticks-value
            TimeTicks,
        arbitrary-value
            Opaque,
        big-counter-value
            Counter64,
        unsigned-integer-value -- includes Gauge32
            Unsigned32
    }

-- in network-byte order

-- (this is a tagged type for historical reasons)
```

```

IpAddress ::=
    [APPLICATION 0]
        IMPLICIT OCTET STRING (SIZE (4))

-- this wraps
Counter32 ::=
    [APPLICATION 1]
        IMPLICIT INTEGER (0..4294967295)

-- this doesn't wrap
Gauge32 ::=
    [APPLICATION 2]
        IMPLICIT INTEGER (0..4294967295)

-- an unsigned 32-bit quantity
-- indistinguishable from Gauge32
Unsigned32 ::=
    [APPLICATION 2]
        IMPLICIT INTEGER (0..4294967295)

-- hundredths of seconds since an epoch
TimeTicks ::=
    [APPLICATION 3]
        IMPLICIT INTEGER (0..4294967295)

-- for backward-compatibility only
Opaque ::=
    [APPLICATION 4]
        IMPLICIT OCTET STRING

-- for counters that wrap in less than one hour with only 32 bits
Counter64 ::=
    [APPLICATION 6]
        IMPLICIT INTEGER (0..18446744073709551615)

-- definition for objects

OBJECT-TYPE MACRO ::=
BEGIN
    TYPE NOTATION ::=
        "SYNTAX" Syntax
        UnitsPart
        "MAX-ACCESS" Access
        "STATUS" Status
        "DESCRIPTION" Text
        ReferPart

        IndexPart
        DefValPart

    VALUE NOTATION ::=
        value(VALUE ObjectName)

    Syntax ::= -- Must be one of the following:
        -- a base type (or its refinement),
        -- a textual convention (or its refinement), or
        -- a BITS pseudo-type
        type
        | "BITS" "{" NamedBits "}"

```



```

NamedBits ::= NamedBit
            | NamedBits "," NamedBit

NamedBit ::= identifier "(" number ")" -- number is nonnegative

UnitsPart ::=
            "UNITS" Text
            | empty

Access ::=
            "not-accessible"
            | "accessible-for-notify"
            | "read-only"
            | "read-write"
            | "read-create"

Status ::=
            "current"
            | "deprecated"
            | "obsolete"

ReferPart ::=
            "REFERENCE" Text
            | empty

IndexPart ::=
            "INDEX"      "{" IndexTypes }"
            | "AUGMENTS" "{" Entry     }"
            | empty

IndexTypes ::=
            IndexType
            | IndexTypes "," IndexType

IndexType ::=
            "IMPLIED" Index
            | Index

Index ::=
            -- use the SYNTAX value of the
            -- correspondent OBJECT-TYPE invocation
            value(ObjectName)

Entry ::=
            -- use the INDEX value of the
            -- correspondent OBJECT-TYPE invocation
            value(ObjectName)

DefValPart ::= "DEFVAL" "{" Defvalue }"
              | empty

Defvalue ::= -- must be valid for the type specified in
             -- SYNTAX clause of same OBJECT-TYPE macro
             value(ObjectSyntax)
             | "{" BitsValue }"

BitsValue ::= BitNames
             | empty

BitNames ::= BitName
            | BitNames "," BitName

BitName ::= identifier
  
```

```

-- a character string as defined in section 3.1.1
Text ::= value(IA5String)
END

-- definitions for notifications

NOTIFICATION-TYPE MACRO ::=
BEGIN
    TYPE NOTATION ::=
        ObjectsPart
        "STATUS" Status
        "DESCRIPTION" Text
        ReferPart

    VALUE NOTATION ::=
        value(VALUE NotificationName)

    ObjectsPart ::=
        "OBJECTS" "{" Objects "}"
    | empty
    Objects ::=
        Object

    Object ::=
    | Objects "," Object
    value(ObjectName)

    Status ::=
        "current"
    | "deprecated"
    | "obsolete"

    ReferPart ::=
        "REFERENCE" Text
    | empty

-- a character string as defined in section 3.1.1
Text ::= value(IA5String)
END

-- definitions of administrative identifiers

zeroDotZero OBJECT-IDENTITY
STATUS      current
DESCRIPTION
    "A value used for null identifiers."
::= { 0 0 }

END

```