

Modelo de información en Internet

Gestión de Redes

Objetivo

2

- Leer e interpretar documentos MIBs
- Es necesario conocer
 - ▣ Sintaxis ASN.1 – trabajo personal
 - ▣ Documentación con las bases: SMI – en clase

1.3.6.1.2.1.4.1

IP-MIB::ipForwarding

SNMPv2-MIB::ipForwarding

Iso.org.dod.internet.mgmt.mib2.ip.ipForwarding

Índice

- Estructura y documentos
- Definiciones – SMI y SMIv2
- Macros
- Ejemplos de declaraciones

Índice

- Estructura y documentos
- Definiciones – SMI y SMIv2
- Macros
- Ejemplos de declaraciones

- **SMI: Structure of Management Information** [RFC1155](#) RFC1155-SMI
 - ▣ Estructura del árbol de gestión.
 - ▣ Declaraciones de tipos utilizados en las MIBs.
 - ▣ Declaración de la macro OBJECT TYPE para definir objetos.
- **Concise MIB Definitions** [RFC1212](#) RFC-1212
 - ▣ Ampliación de la macro de OBJECT TYPE.
- **SMIv2:** [RFC2578](#) SNMPv2-SMI
- Textual Conventions for SMIv2 [RFC2579](#) SNMPv2-TC
- Conformance Statements for SMIv2 [RFC2580](#) SNMPv2-CONF
- **MIB** estándar en [RFC1156](#) y **MIB-2** estándar en [RFC1213](#) RFC1213-MIB



Los más típicos:

MIB estándar en RFC1156

MIB-2 estándar en RFC1213

TCP-MIB: RFC4022

UDP-MIB: RFC4113

IP-MIB: RFC4293

IF-MIB: RFC2863

HostResources: RFC1514 → RFC2790



Referencias

Visor de MIBs:

<http://www.simpleweb.org/ietf/mibs/>

Buscador de RFCs:

<http://tools.ietf.org/html/>

Ejemplos de estructura de MIBs

7

MIBs-II (RFC1213)

- system
- interfaces
- at
- ip
- icmp
- tcp
- udp
- egp
- snmp

HostResources

- hrSystem
- hrStorage
- hrDevice
 - hrProcessor
 - hrNetwork
 - hrPrinter
 - hrDisk
 - hrPartition
 - hrFS
- hrSWRun
- hrSWRunPerf
- hrSWInstalled

```
RFC1155-SMI DEFINITIONS ::= BEGIN
```

```
    EXPORTS -- EVERYTHING
```

```
        internet, directory, mgmt,  
        experimental, private, enterprises,  
        OBJECT-TYPE, ObjectName, ObjectSyntax,  
        SimpleSyntax, ApplicationSyntax,  
        NetworkAddress, IpAddress,  
        Counter, Gauge, TimeTicks, Opaque;
```

```
    .....OTRAS DECLARACIONES.....
```

```
END
```

SMI – Estructura del árbol de gestión

9

Define la estructura del árbol de MIB en ASN.1.

internet OBJECT IDENTIFIER ::= { iso org(3) dod(6) 1 }

directory OBJECT IDENTIFIER ::= { internet 1 }

mgmt OBJECT IDENTIFIER ::= { internet 2 }

experimental OBJECT IDENTIFIER ::= { internet 3 }

private OBJECT IDENTIFIER ::= { internet 4 }

enterprises OBJECT IDENTIFIER ::= { private 1 }

SNMPv2-SMI DEFINITIONS ::= BEGIN

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 }

SMIPv2 (Domains and time)

11

-- transport domains

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

snmpProxys OBJECT IDENTIFIER ::= { snmpV2 2 } -- module identities

snmpModules OBJECT IDENTIFIER ::= { snmpV2 3 }

--Extended UTCTime

ExtUTCTime ::= OCTET STRING(SIZE(11 | 13))

-- format is YYMMDDHHMMZ or YYYYMMDDHHMMZ

Índice

- Estructura y documentos
- **Definiciones – SMI y SMIv2**
- Macros
- Ejemplos de declaraciones



Está permitido usar los siguientes tipos universales:

Primitivos: **INTEGER**, **OCTET STRING**, **NULL**, **OBJECT IDENTIFIER**.

Construidos: **SEQUENCE** y **SEQUENCE OF**.



Los enumerados se declaran como enteros y no se utiliza el valor 0.

- Tipo predefinidos:
 - ▣ **NetworkAddress**: CHOICE que permite seleccionar varios formatos de direcciones. Actualmente sólo IpAddress.
 - ▣ **IpAddress**: Dirección IP versión 4.
 - ▣ **Counter**: Entero no negativo de 32 bits que puede ser incrementado pero no decrementado. Cuando alcanza el valor máximo comienza por 0. Como alternativa se puede utilizar un latch Counter.
 - ▣ **Gauge**: Entero no negativo de 32 bits que puede ser incrementado o decrementado. Cuando alcanza el valor máximo permanece con ese valor hasta que se pone a cero.
 - ▣ **TimeTicks**: Entero no negativo que cuenta tiempo en centésimas de segundo (tiempo relativo).
 - ▣ **Opaque**: Datos arbitrarios codificados como OCTET STRING.

Declaración de los tipos definidos

15

```
-- application-wide types
```

```
NetworkAddress ::= CHOICE {internet IpAddress}
```

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

```
Counter ::= [APPLICATION 1] IMPLICIT INTEGER (0..4294967295)
```

```
Gauge ::= [APPLICATION 2] IMPLICIT INTEGER (0..4294967295)
```

```
TimeTicks ::= [APPLICATION 3] IMPLICIT INTEGER (0..4294967295)
```

```
Opaque ::= [APPLICATION 4] IMPLICIT OCTET STRING
```

SMlv2:Syntax

16

ObjectName ::= OBJECT IDENTIFIER

NotificationName ::= OBJECT IDENTIFIER

ObjectSyntax ::=

CHOICE {

simple SimpleSyntax,

application-wide ApplicationSyntax

}

SimpleSyntax ::=

CHOICE {

integer-value INTEGER (-2147483648..2147483647),

string-value OCTET STRING (SIZE (0..65535)),

objectID-value OBJECT IDENTIFIER

}

Integer32 ::= INTEGER (-2147483648..2147483647)

SMIPv2:Syntax

17

ApplicationSyntax ::=

CHOICE {

ipAddress-value IpAddress,

counter-value Counter32,

timeticks-value TimeTicks,

arbitrary-value Opaque,

big-counter-value Counter64,

unsigned-integer-value Unsigned32 -- includes Gauge32 }

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

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

Gauge32 ::= [APPLICATION 2] IMPLICIT INTEGER (0..4294967295)

Unsigned32 ::= [APPLICATION 3] IMPLICIT INTEGER (0..4294967295)

TimeTicks ::= [APPLICATION 4] IMPLICIT INTEGER (0..4294967295)

Opaque ::= [APPLICATION 5] IMPLICIT OCTET STRING

Counter64 ::= [APPLICATION 6] IMPLICIT INTEGER (0..18446744073709551615)

zeroDotZero OBJECT-IDENTITY

STATUS current

DESCRIPTION "A value used for null identifiers."

::= { 0 0 }

Índice

- Estructura y documentos
- Definiciones – SMI y SMIv2
- **Macros**
- Ejemplos de declaraciones

Declaración de objetos

20

- Utiliza la macro de definición de objetos (originalmente definida en SMI RFC1155 y ampliada en Concise MIB Definitions RFC1212):

```
IMPORTS  ObjectName,ObjectSyntax  FROM RFC1155-SMI
         DisplayString FROM RFC1158-MIB;

OBJECT-TYPE MACRO ::=
BEGIN
.....ver siguiente diapositiva.....
END
IndexSyntax ::= CHOICE {
    number INTEGER (0..MAX),
    string OCTET STRING,
    object OBJECT IDENTIFIER,
    address NetworkAddress,
    ipAddress IpAddress
}
```

DisplayString OCTET STRING

Declaración de objetos: macro OBJECT TYPE

21

```
OBJECT-TYPE MACRO ::=
BEGIN
  TYPE NOTATION ::=
    "SYNTAX" type(ObjectSyntax)
    "ACCESS" Access
    "STATUS" Status
    DescrPart
    ReferPart
    IndexPart
    DefValPart

  VALUE NOTATION ::= value (VALUE ObjectName)
  Access ::= "read-only" | "read-write" | "write-only" | "not-accessible"
  Status ::= "mandatory" | "optional" | "obsolete" | "deprecated"
  DescrPart ::= "DESCRIPTION" value (description DisplayString) | empty
  ReferPart ::= "REFERENCE" value (reference DisplayString) | empty
  IndexPart ::= "INDEX" "{" IndexTypes "}" | empty
  IndexTypes ::= IndexType | IndexTypes "," IndexType
  IndexType ::= value (indexobject ObjectName) | type (indextype) --IndexSyntax
  DefValPart ::= "DEFVAL" "{" value (defvalue ObjectSyntax) "}" | empty
END
```

Declaración de objetos

22

```
ObjectName ::= OBJECT IDENTIFIER
```

```
Objectsyntax ::= CHOICE {simple SimpleSyntax,  
                           application-wide ApplicationSyntax}
```

```
SimpleSyntax ::= CHOICE {number INTEGER, string OCTET STRING,  
                           object OBJECT IDENTIFIER, empty NULL}
```

```
ApplicationSyntax ::= CHOICE {address NetworkAddress, counter Counter,  
                               gauge Gauge, ticks TimeTicks, arbitrary Opaque }
```

Declaración de objetos

23

- ❑ SYNTAX: sintaxis de la instancia del objeto.
- ❑ ACCESS: formas en que una instancia de un objeto puede ser accedida (read-only, read-write, write-only, not-accesible).
- ❑ STATUS: requisitos de implementación:
 - ❑ Mandatory: obligatorio.
 - ❑ Optional: opcional.
 - ❑ Deprecated: debe ser soportado, pero será quitado en la siguiente versión de MIBs.
 - ❑ Obsolete: ya no es necesario implementarlo.
- ❑ DescrPart: (opcional) descripción textual de la semántica del objeto.
- ❑ ReferPart: (opcional) referencia cruzada textual a un objeto definido en otra MIB.
- ❑ IndexPart: define índices de las tablas.
- ❑ DefValPart: (opcional) valor por defecto usado cuando se crea una instancia del objeto.
- ❑ VALUE NOTATION: OID asignado al objeto.

SMIv2:OBJECT TYPE MACRO

24

OBJECT-TYPE MACRO ::=

BEGIN

TYPE NOTATION ::=

"SYNTAX" Syntax UnitsPart

"MAX-ACCESS" Access

"STATUS" Status

"DESCRIPTION" Text

ReferPart

IndexPart

DefValPart

VALUE NOTATION ::= value(VALUE ObjectName)

END

Declarados en la siguiente transparencia

SMIv2:OBJECT TYPE MACRO

25

```
Syntax ::= type | "BITS" "{" NamedBits "}"
NamedBits ::= NamedBit | NamedBits "," NamedBit
NamedBit ::= identifier "(" number ")"
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 ::= value(ObjectName)
Entry ::= value(ObjectName)
DefValPart ::= "DEFVAL" "{" Defvalue "}" | empty
Defvalue ::= value(ObjectSyntax) | "{" BitsValue "}"
BitsValue ::= BitNames | empty
BitNames ::= BitName | BitNames "," BitName
BitName ::= identifier
Text ::= value(IA5String)
```

Otras Macros definidas en SMIv2

26

MODULE-IDENTITY

comunica información y semántica de un módulo

OBJECT-IDENTITY define información sobre OIDs administrativos

NOTIFICATION-TYPE define información de notificación

TEXTUAL-CONVENTION

comunica la sintaxis y semántica asociada a una convención textual

OBJECT-GROUP define grupos de objetos relacionados

NOTIFICATION-GROUP define grupos de notificaciones

MODULE-COMPLIANCE

informa sobre los requisitos mínimos para implementar un módulo MIB.

AGENT-CAPABILITIES

informa sobre las capacidades de una entidad agente SNMPv2

SMIv2: MODULE-IDENTITY MACRO

27

```
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
  Text ::= value(IA5String)
END
```

Uah-tel-p2ptls MODULE-IDENTITY

LAST-UPDATED "9505241813Z"

ORGANIZATION "Alcala University"

CONTACT-INFO "xxx@uah.es"

DESCRIPTION "The MIB module for entities implementing the p2ptls protocol developed in UAH."

REVISION "9505241813Z"

DESCRIPTION "The latest version of this MIB module."

REVISION "9210070433Z"

DESCRIPTION "The initial version of this MIB module."

::= { experimental xx }

SMIv2: OBJECT-IDENTITY MACRO

29

```
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
```

```
    Text ::= value(IA5String)
```

```
END
```

Ejemplo

30

uah-tel OBJECT-IDENTITY

STATUS current

DESCRIPTION "The authoritative identity of the
Telematic group in UAH."

::= { uah 1 }

SMIv2: NOTIFICATION-TYPE MACRO

31

```
NOTIFICATION-TYPE MACRO ::=
BEGIN
  TYPE NOTATION ::=
    ObjectsPart
    "STATUS" Status
    "DESCRIPTION" Text ReferPart
  VALUE NOTATION ::= value(VALUE NotificationName)
  ObjectsPart ::= "OBJECTS" "{" Objects "}" | empty
  Objects ::= Object | Objects "," Object
  Object ::= value(ObjectName)
  Status ::= "current" | "deprecated" | "obsolete"
  ReferPart ::= "REFERENCE" Text | empty
  Text ::= value(IA5String)
END
```

linkUp NOTIFICATION-TYPE

OBJECTS { ifIndex }

STATUS current

DESCRIPTION "A linkUp trap signifies that the SNMPv2 entity, acting in an agent role, recognizes that one of the communication links represented in its configuration has come up."

::= { snmpTraps 4 }

TEXTUAL-CONVENTION for SMIv2 (RFC2579)

33

```
TEXTUAL-CONVENTION MACRO ::=BEGIN
    TYPE NOTATION ::= DisplayPart
        "STATUS" Status
    "DESCRIPTION" Text
        ReferPart
        "SYNTAX" Syntax
    VALUE NOTATION ::= value(VALUE Syntax)    -- adapted ASN.1
    DisplayPart ::= "DISPLAY-HINT" Text | empty
    Status ::= "current" | "deprecated" | "obsolete"
    ReferPart ::= "REFERENCE" Text | empty
    Syntax ::= type | "BITS" "{" NamedBits "}"
    NamedBits ::= NamedBit | NamedBits "," NamedBit
    NamedBit ::= identifier "(" number ")" -- number is nonnegative
END
```

TEXTUAL-CONVENTION for SMIPv2 (RFC2579)

34

Declaraciones con la macro:

DisplayString

PhysAddress

MacAddress

TruthValue

TestAndIncr

AutonomousType

InstancePointer

VariablePointer

RowPointer

RowStatus

```
PhysAddress ::= TEXTUAL-CONVENTION
```

```
    DISPLAY-HINT "1 x:"
```

```
    STATUS      current
```

```
    DESCRIPTION
```

```
        "Represents media- or physical-level addresses."
```

```
SYNTAX      OCTET STRING
```

1 x: indica el formato de representación del octet string

Cada octeto se visualiza en hexadecimal separado por “.”

OBJECT GROUP MACRO (RFC2580)

35

- Estructura
 - ▣ ObjectParts: secuencia de elementos de tipo ObjectName
 - ▣ STATUS: current, deprecated, obsolete
 - ▣ DESCRIPTION: Text
 - ▣ ReferPart: REFERENCE Text | empty
- Ver declaraciones en <http://tools.ietf.org/html/rfc2580>

```
snmpGroup OBJECT-GROUP
    OBJECTS {
        snmplnPks,
        snmplnBadVersions,
        snmplnASNParseErrs,
        snmpBadOperations,
        snmpSilentDrops,
        snmpProxyDrops,
        snmpEnableAuthenTraps }

STATUS current

DESCRIPTION "A collection of objects
    providing basic instrumentation and
    control of an SNMPv2 entity."

::= { snmpMIBGroups 8 }
```

NOTIFICATION-GROUP MACRO (RFC2580)

36

- Estructura:
 - ▣ NotificationParts: secuencia de elementos de tipo NotificationName
 - ▣ STATUS: current, deprecated, obsolete
 - ▣ DESCRIPTION: Text
 - ▣ ReferPart: REFERENCE Text | empty
- Ver declaraciones en <http://tools.ietf.org/html/rfc2580>

```
snmpBasicNotificationsGroup NOTIFICATION-GROUP
```

```
    NOTIFICATIONS { coldStart, authenticationFailure }
```

```
    STATUS        current
```

```
    DESCRIPTION   "The two notifications which an SNMPv2 entity is required to  
implement."
```

```
::= { snmpMIBGroups 7 }
```

MODULE-COMPLIANCE (RFC2580)

37

□ Estructura:

- STATUS: current, deprecated, obsolete
- DESCRIPTION: Text
- ReferPart: REFERENCE Text | empty
- ModulePart: sequence of
 - ModuleName: ModuleIdentifier (OID) | empty,
 - MandatoryPart: sequence of groups (OID) | empty
 - CompliancePart: sequence of ComplianceGroup {OID, DESCRIPTION Text} | Object (ObjectName, SyntaxPart, WriteSyntaxPart, AccessPart)

telMIB Compliance MODULE-COMPLIANCE

STATUS current

DESCRIPTION "The compliance statement for TELv2 entities which implement the TELv2 MIB."

MODULE -- compliance to the containing MIB module

MANDATORY-GROUPS { telSystemGroup, telStatsGroup, telTrapGroup, telSetGroup, telNotificationsGroup }

GROUP telV1Group

DESCRIPTION "The telV1 group is mandatory only for those TELv2 entities which also implement TELv1."

::= { telMIBCompliances 1 }

AGENT-CAPABILITIES (RFC2580)

38

- Estructura:
 - ▣ PRODUCT-RELEASE Text
 - ▣ STATUS: current, deprecated, obsolete
 - ▣ DESCRIPTION: Text
 - ▣ ReferPart: REFERENCE Text | empty
 - ▣ ModulePart empty | Sequence of Modules (Name, Groups, Variations))

exampleAgent AGENT-CAPABILITIES

PRODUCT-RELEASE "ACME Agent release 1.1 for 4BSD"

STATUS current

DESCRIPTION "ACME agent for 4BSD"

SUPPORTS SNMPv2-MIB

INCLUDES { systemGroup, snmpGroup, snmpSetGroup, snmpBasicNotificationsGroup }

VARIATION coldStart

DESCRIPTION "A coldStart trap is generated on all reboots."

AGENT-CAPABILITIES (RFC2580)

39

```
SUPPORTS      IF-MIB

INCLUDES      { ifGeneralGroup, ifPacketGroup }

VARIATION     ifAdminStatus

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

              DESCRIPTION  "Unable to set test mode on 4BSD"

VARIATION     ifOperStatus

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

              DESCRIPTION  "Information limited on 4BSD"

SUPPORTS      IP-MIB

*****

SUPPORTS      TCP-MIB

*****

SUPPORTS      UDP-MIB

*****

::= { acmeAgents 1 }
```

Índice

- Estructura y documentos
- Definiciones – SMI y SMIv2
- Macros
- Ejemplos de declaraciones

Ejemplos de declaraciones

41

```
RFC1213-MIB DEFINITIONS ::= BEGIN

    IMPORTS mgmt, NetworkAddress, IpAddress, Counter, Gauge,
            TimeTicks FROM RFC1155-SMI
            OBJECT-TYPE FROM RFC-1212;

    PhysAddress ::= OCTET STRING
    DisplayString ::= OCTET STRING --size(0..255)
    mib-2 OBJECT IDENTIFIER ::= { mgmt 1 }      -- MIB-II
    system      OBJECT IDENTIFIER ::= { mib-2 1 }
    interfaces  OBJECT IDENTIFIER ::= { mib-2 2 }
    at          OBJECT IDENTIFIER ::= { mib-2 3 }
    ip          OBJECT IDENTIFIER ::= { mib-2 4 }
    icmp        OBJECT IDENTIFIER ::= { mib-2 5 }
    tcp         OBJECT IDENTIFIER ::= { mib-2 6 }
    udp         OBJECT IDENTIFIER ::= { mib-2 7 }
    egp         OBJECT IDENTIFIER ::= { mib-2 8 }
    -- cmot     OBJECT IDENTIFIER ::= { mib-2 9 }
    transmission OBJECT IDENTIFIER ::= { mib-2 10 }
    snmp        OBJECT IDENTIFIER ::= { mib-2 11 }
```

Ejemplos de declaraciones

42

`sysDescr OBJECT-TYPE`

`SYNTAX DisplayString (SIZE (0..255))`

`ACCESS read-only`

`STATUS mandatory`

`DESCRIPTION`

"A textual description of the entity. This value should include the full name and version identification of the system's hardware type, software operating-system, and networking software. It is mandatory that this only contain printable ASCII characters."

`::= { system 1 }`

Ejemplos de declaraciones

43

```
ifNumber OBJECT-TYPE
```

```
    SYNTAX  INTEGER
```

```
    ACCESS  read-only
```

```
    STATUS  mandatory
```

```
    DESCRIPTION
```

```
        "The number of network interfaces (regardless of  
         their current state) present on this system."
```

```
 ::= { interfaces 1 }
```

Ejemplos de declaraciones

44

ipForwarding OBJECT-TYPE

SYNTAX INTEGER {

forwarding(1), -- acting as a gateway

not-forwarding(2) -- NOT acting as a gateway

}

ACCESS read-write

STATUS mandatory

DESCRIPTION

"The indication of whether this entity is acting as an IP gateway in respect to the forwarding of datagrams received by, but not addressed to, this entity. IP gateways forward datagrams. IP hosts do not (except those source-routed via the host). Note that for some managed nodes, this object may take on only a subset of the values possible. Accordingly, it is appropriate for an agent to return a 'badValue' response if a management station attempts to change this object to an inappropriate value."

::= { ip 1 }

- Se declara:
 - ▣ Nombre de la tabla: Table
 - ▣ Entry: tableEntry
 - ▣ Los elementos columnares.
 - ▣ El índice.
- Cada tabla consiste en SEQUENCE OF TableEntry.
- Cada fila (TableEntry) consiste en una SEQUENCE de elementos columnares.

Ejemplo de tabla: ipRoutingTable

46

Tabla

```
ipRouteTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF IpRouteEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "This entity's IP Routing table."
    ::= { ip 21 }
```

Entry y declaración
del Índice

```
ipRouteEntry OBJECT-TYPE
    SYNTAX  IpRouteEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "A route to a particular destination."
    INDEX   { ipRouteDest }
    ::= { ipRouteTable 1 }
```

Ejemplo de tabla: ipRoutingTable

47

Objetos columnares

```
IpRouteEntry ::= SEQUENCE {  
    ipRouteDest IpAddress,  
    ipRouteIfIndex INTEGER,  
    ipRouteMetric1 INTEGER,  
    ipRouteMetric2 INTEGER,  
    ipRouteMetric3 INTEGER,  
    ipRouteMetric4 INTEGER,  
    ipRouteNextHop IpAddress,  
    ipRouteType      INTEGER,  
    ipRouteProto      INTEGER,  
    ipRouteAge        INTEGER,  
    ipRouteMask       IpAddress  
    ipRouteMetric5 INTEGER,  
    ipRouteInfo OBJECT IDENTIFIER  
}
```

Ejemplo de tabla: ipRoutingTable

48

ipRouteDest OBJECT-TYPE

SYNTAX IpAddress

ACCESS read-write

STATUS mandatory

::= { ipRouteEntry 1 }

ipRouteIfIndex OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-write

STATUS mandatory

::= { ipRouteEntry 2 }

ipRouteMetric1 OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-write

STATUS mandatory

::= { ipRouteEntry 3 }

ipRouteMetric2 SYNTAX INTEGER

ACCESS read-write

STATUS mandatory

::= { ipRouteEntry 4 }

ipRouteMetric3 OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-write

STATUS mandatory

::= { ipRouteEntry 5 }

ipRouteMetric4 OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-write

STATUS mandatory

::= { ipRouteEntry 6 }

Ejemplo de tabla: ipRoutingTable

49

ipRouteNextHop OBJECT-TYPE

SYNTAX IpAddress

ACCESS read-write

STATUS mandatory

::= { ipRouteEntry 7 }

ipRouteType OBJECT-TYPE

SYNTAX INTEGER {

other(1), -- none of the following

invalid(2), -- an invalidated route route to directly

direct(3), -- connected (sub-)network route to a non-local

remote(4) -- host/network/sub-network

}

ACCESS read-write

STATUS mandatory

::= { ipRouteEntry 8 }

Ejemplo de tabla: ipRoutingTable

50

ipRouteProto OBJECT-TYPE

SYNTAX INTEGER {

other(1), -- none of the following non-protocol information
 -- e.g., manually

local(2), -- configured entries set via a network

netmgmt(3), -- management protocol obtained via ICMP,

icmp(4), -- e.g., Redirect the following are gateway routing
 -- protocols

egp(5),

ggp(6),

hello(7),

rip(8),

is-is(9),

es-is(10),

ciscoIgrp(11),

bbnSpfIgp(12),

ospf(13)

bgp(14)}

ACCESS read-only

STATUS mandatory

::= { ipRouteEntry 9 }

Ejemplo de tabla: ipRoutingTable

51

ipRouteAge OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-write

STATUS mandatory

::= { ipRouteEntry 10 }

ipRouteMask OBJECT-TYPE

SYNTAX IpAddress

ACCESS read-write

STATUS mandatory

::= { ipRouteEntry 11 }

ipRouteInfo OBJECT-TYPE

SYNTAX OBJECT IDENTIFIER

ACCESS read-only

STATUS mandatory

::= { ipRouteEntry 13 }

ipRouteMetric5 OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-write

STATUS mandatory

::= { ipRouteEntry 12 }

Ejemplo de tabla: tcpConnTable

52

tcpConnTable OBJECT-TYPE

SYNTAX SEQUENCE OF TcpConnEntry

ACCESS not-accessible

STATUS mandatory

::= { tcp 13 }

tcpConnEntry OBJECT-TYPE

SYNTAX TcpConnEntry

ACCESS not-accessible

STATUS mandatory

INDEX { tcpConnLocalAddress,
tcpConnLocalPort,
tcpConnRemAddress,
tcpConnRemPort }

::= { tcpConnTable 1 }

TcpConnEntry ::= SEQUENCE {
tcpConnState INTEGER,
tcpConnLocalAddress IpAddress,
tcpConnLocalPort INTEGER (0..65535),
tcpConnRemAddress IpAddress,
tcpConnRemPort INTEGER (0..65535)}

Ejemplo de tabla: tcpConnTable

53

tcpConnState OBJECT-TYPE

```
SYNTAX  INTEGER {
    closed(1),
    listen(2),
    synSent(3),
    synReceived(4),
    established(5),
    finwait1(6),
    finwait2(7),
    closewait(8),
    lastAck(9),
    closing(10),
    timewait(11),
    deleteTCB(12)
}
```

ACCESS read-write

STATUS mandatory

::= { tcpConnEntry 1 }

tcpConnLocalAddress OBJECT-TYPE

SYNTAX IPAddress

ACCESS read-only

STATUS mandatory

::= { tcpConnEntry 2 }

tcpConnLocalPort OBJECT-TYPE

SYNTAX INTEGER (0..65535)

ACCESS read-only

STATUS mandatory

::= { tcpConnEntry 3 }

Ejemplo de tabla: tcpConnTable

54

tcpConnRemAddress OBJECT-TYPE

SYNTAX IpAddress

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The remote IP address for this TCP connection."

::= { tcpConnEntry 4 }

tcpConnRemPort OBJECT-TYPE

SYNTAX INTEGER (0..65535)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The remote port number for this TCP connection."

::= { tcpConnEntry 5 }

Ejemplos de declaraciones

55

- Los módulos MIB se van actualizando con el tiempo
 - ▣ RFC1213 → ip (junto a otros protocolos) – Año 1991
 - ▣ RFC4293 → ip (diferencia IPv4 e IPv6) – Año 2006