```
RFC1155-SMI DEFINITIONS ::= BEGIN

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

    -- the path to the root

    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 }

    -- definition of object types

    OBJECT-TYPE MACRO ::=
    BEGIN
      TYPE NOTATION ::= "SYNTAX" type (TYPE ObjectSyntax)
                  "ACCESS" Access
                  "STATUS" Status
      VALUE NOTATION ::= value (VALUE ObjectName)

      Access ::= "read-only"
              | "read-write"
              | "write-only"
              | "not-accessible"
      Status ::= "mandatory"
              | "optional"
              | "obsolete"
    END

      -- names of objects in the MIB

      ObjectName ::=
        OBJECT IDENTIFIER

      -- syntax of objects in the MIB

      ObjectSyntax ::=
        CHOICE {
          simple
            SimpleSyntax,

      -- note that simple SEQUENCEs are not directly mentioned here to keep things simple (i.e.,
      -- prevent misuse).  However, application-wide types which are IMPLICITLY encoded simple
      -- SEQUENCEs may appear in the following CHOICE

          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

-- other application-wide types, as they are
-- defined, will be added here
   }

-- application-wide types

NetworkAddress ::=
   CHOICE {
      internet     IpAddress
   }

IpAddress ::=
   [APPLICATION 0]        -- in network-byte order
      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]        -- arbitrary ASN.1 value,
      IMPLICIT OCTET STRING

END
```

RFC1213-MIB

```
DEFINITIONS ::= BEGIN
 IMPORTS
        mgmt, NetworkAddress, IpAddress, Counter, Gauge, TimeTicks
                FROM RFC1155-SMI
        OBJECT-TYPE
                FROM RFC-1212;

--  MIB-II (same prefix as MIB-I)
mib-2      OBJECT IDENTIFIER ::= { mgmt 1 }

-- textual conventions
DisplayString ::= OCTET STRING
-- This data type is used to model textual information taken
-- from the NVT ASCII character set.  By convention, objects
-- with this syntax are declared as having SIZE (0..255)

PhysAddress ::= OCTET STRING
-- This data type is used to model media addresses.  For many
-- types of media, this will be in a binary representation.
-- For example, an ethernet address would be represented as a string of 6 octets.

-- groups in 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 }
-- historical (some say hysterical)
-- cmot         OBJECT IDENTIFIER ::= { mib-2 9 }
transmission    OBJECT IDENTIFIER ::= { mib-2 10 }
snmp            OBJECT IDENTIFIER ::= { mib-2 11 }

-- the System group
-- Implementation of the System group is mandatory for all systems.  If an agent is not configured
-- to have a value for any of these variables, a string of length 0 is returned.

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 }

sysObjectID OBJECT-TYPE
    SYNTAX  OBJECT IDENTIFIER
    ACCESS  read-only
    STATUS  mandatory
```

```
   DESCRIPTION       "The vendor's authoritative identification of the network management
          subsystem contained in the entity.  This value is allocated within the SMI enterprises
          subtree (1.3.6.1.4.1) and provides an easy and unambiguous means for determining `what
          kind of box' is being managed.  For example, if vendor `Flintstones, Inc.' was assigned the
          subtree 1.3.6.1.4.1.4242, it could assign the identifier 1.3.6.1.4.1.4242.1.1 to its
          `FredRouter'."
   ::= { system 2 }

sysUpTime OBJECT-TYPE
   SYNTAX  TimeTicks
   ACCESS  read-only
   STATUS  mandatory
   DESCRIPTION       "The time (in hundredths of a second) since the network management
          portion of the system was last re-initialized."
   ::= { system 3 }

sysContact OBJECT-TYPE
   SYNTAX  DisplayString (SIZE (0..255))
   ACCESS  read-write
   STATUS  mandatory
   DESCRIPTION       "The textual identification of the contact person for this managed node,
          together with information on how to contact this person."
   ::= { system 4 }

sysName OBJECT-TYPE
   SYNTAX  DisplayString (SIZE (0..255))
………
-- the Interfaces group

-- Implementation of the Interfaces group is mandatory for all systems.
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 }

-- the Interfaces table
-- The Interfaces table contains information on the entity's interfaces.  Each interface is thought
-- of as being attached to a `subnetwork'.  Note that this term should not be confused with
-- `subnet' which refers to an addressing partitioning scheme used in the Internet suite of protocols.
ifTable OBJECT-TYPE
   SYNTAX  SEQUENCE OF IfEntry
   ACCESS  not-accessible
   STATUS  mandatory
   DESCRIPTION       "A list of interface entries.  The number of entries is given by the value of
          ifNumber."
   ::= { interfaces 2 }

ifEntry OBJECT-TYPE
   SYNTAX  IfEntry
   ACCESS  not-accessible
   STATUS  mandatory
   DESCRIPTION      "An interface entry containing objects at the subnetwork layer and below
          for a particular interface."
   INDEX   { ifIndex }
   ::= { ifTable 1 }

IfEntry ::=
   SEQUENCE {
      ifIndex              INTEGER,          ifDescr              DisplayString,
      ifType               INTEGER,          ifMtu                INTEGER,
      ifSpeed              Gauge,            ifPhysAddress        PhysAddress,
      ifAdminStatus        INTEGER,          ifOperStatus         INTEGER,
      ifLastChange         TimeTicks,        ifInOctets           Counter,
      ifInUcastPkts        Counter,          ifInNUcastPkts       Counter,
      ifInDiscards         Counter,          ifInErrors           Counter,
      ifInUnknownProtos    Counter,          ifOutOctets          Counter,
      ifOutUcastPkts       Counter,          ifOutNUcastPkts      Counter,
      ifOutDiscards        Counter,          ifOutErrors          Counter,
      ifOutQLen            Gauge,            ifSpecific           OBJECT IDENTIFIER
}

ifIndex OBJECT-TYPE
   SYNTAX  INTEGER
   ACCESS  read-only
   STATUS  mandatory
   DESCRIPTION      "A unique value for each interface.  Its value ranges between 1 and the
          value of ifNumber.  The value for each interface must remain constant at least from one re-
          initialization of the entity's network management system to the next re-initialization."
   ::= { ifEntry 1 }

ifDescr OBJECT-TYPE
   SYNTAX  DisplayString (SIZE (0..255))
   ACCESS  read-only
   STATUS  mandatory
   DESCRIPTION      "A textual string containing information about the interface.  This string
          should include the name of the manufacturer, the product name and the version of the
          hardware interface."
   ::= { ifEntry 2 }
```

………

Exemples d'instance et de valeurs (obtenues par snmpwalk)

```
ifNumber.0 = .1.3.6.1.2.1.2.1.0 = INTEGER: 3
ifIndex.1 = .1.3.6.1.2.1.2.2.1.1.1 = INTEGER: 1
ifIndex.2 = .1.3.6.1.2.1.2.2.1.1.2 = INTEGER: 2
ifIndex.3 = .1.3.6.1.2.1.2.2.1.1.3 = INTEGER: 3
ifDescr.1 = .1.3.6.1.2.1.2.2.1.2.1 = STRING: lo
ifDescr.2 = .1.3.6.1.2.1.2.2.1.2.2 = STRING: eth0
ifDescr.3 = .1.3.6.1.2.1.2.2.1.2.3 = STRING: sit0
ifType.1 = .1.3.6.1.2.1.2.2.1.3.1 = INTEGER: softwareLoopback(24)
ifType.2 = .1.3.6.1.2.1.2.2.1.3.2 = INTEGER: ethernetCsmacd(6)
ifType.3 = .1.3.6.1.2.1.2.2.1.3.3 = INTEGER: tunnel(131)
ifMtu.1 = .1.3.6.1.2.1.2.2.1.4.1 = INTEGER: 16436
ifMtu.2 = .1.3.6.1.2.1.2.2.1.4.2 = INTEGER: 1500
ifMtu.3 = .1.3.6.1.2.1.2.2.1.4.3 = INTEGER: 1480
ifSpeed.1 = .1.3.6.1.2.1.2.2.1.5.1 = Gauge32: 10000000
ifSpeed.2 = .1.3.6.1.2.1.2.2.1.5.2 = Gauge32: 100000000
ifSpeed.3 = .1.3.6.1.2.1.2.2.1.5.3 = Gauge32: 0
…
```