

Username: Fachhochschule Augsburg **Book:** Essential SNMP, 2nd Edition. No part of any chapter or book may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher of the book or chapter. Redistribution or other use that violates the fair use privilege under U.S. copyright laws (see 17 USC107) or that otherwise violates these Terms of Service is strictly prohibited. Violators will be prosecuted to the full extent of U.S. Federal and Massachusetts laws.

Extensions to the SMI in Version 2

SMIv2 extends the SMI object tree by adding the *snmpV2* branch to the *internet* subtree, adding several new datatypes and making a number of other changes. Figure 2-3 shows how the *snmpV2* objects fit into the bigger picture; the OID for this new branch is 1.3.6.1.6.3.1.1, or *iso.org.dod.internet.snmpV2.snmpModules.snmpMIB.snmpMIBObjects*. SMIv2 also defines some new datatypes, which are summarized in Table 2-2.

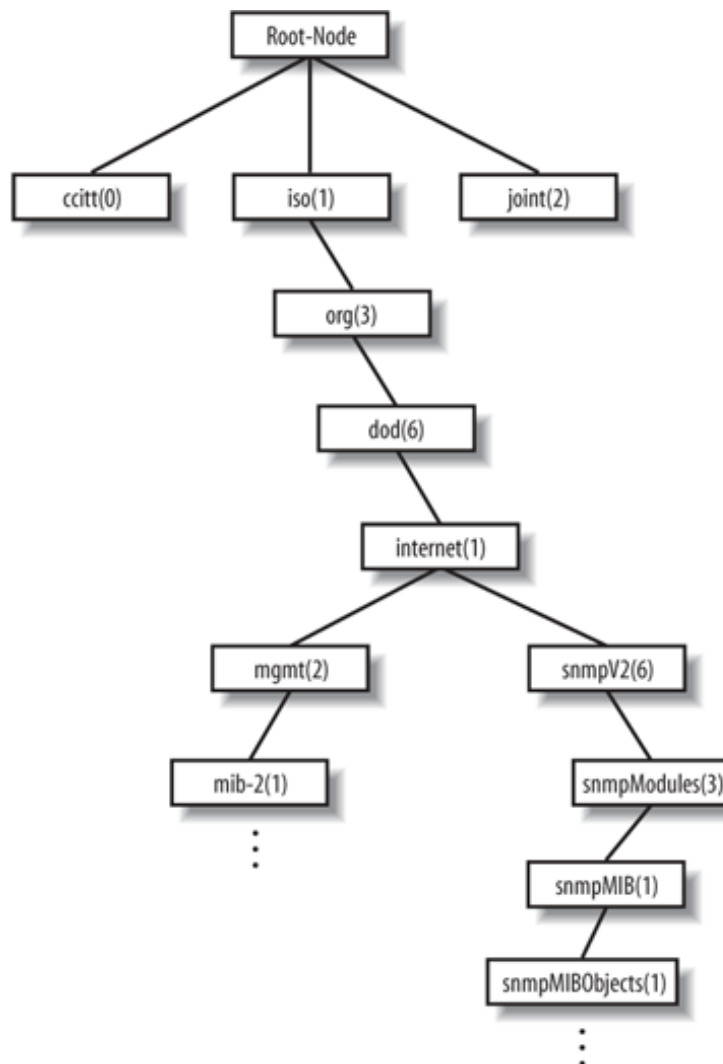


Figure 2-3. SMIv2 registration tree for SNMPv2

Table 2-2. New datatypes for SMiv2

Datatype	Description
Integer32	Same as an INTEGER.
Counter32	Same as a Counter.
Gauge32	Same as a Gauge.
Unsigned32	Represents decimal values in the range of 0 to $2^{32} - 1$, inclusive.
Counter64	Similar to Counter32, but its maximum value is 18,446,744,073,709,551,615. Counter64 is ideal for situations in which a Counter32 may wrap back to 0 in a short amount of time.
BITS	An enumeration of nonnegative named bits.

The definition of an object in SMiv2 has changed slightly from SMiv1. There are some new optional fields, giving you more control over how an object is accessed, allowing you to augment a table by adding more columns, and letting you give better descriptions. Here's the syntax of an object definition for SMiv2. The changed parts are in bold:

```

<name> OBJECT-TYPE
    SYNTAX <datatype>
    UnitsParts <Optional, see below>
    MAX-ACCESS <See below>
    STATUS <See below>
    DESCRIPTION
        "Textual description describing this particular managed object."
    AUGMENTS { <name of table> }
    ::= { <Unique OID that defines this object> }

```

Table 2-3 briefly describes the object definition enhancements made in SMiv2.

Table 2-3. SMIPv2 object definition enhancements

Object definition enhancement	Description
UnitsParts	A textual description of the units (i.e., seconds, milliseconds, etc.) used to represent the object.
MAX-ACCESS	An OBJECT-TYPE's ACCESS can be MAX-ACCESS in SNMPv2. The valid options for MAX-ACCESS are read-only, read-write, read-create, not-accessible, and accessible-for-notify.
STATUS	This clause has been extended to allow the current, obsolete, and deprecated keywords. current in SNMPv2 is the same as mandatory in an SNMPv1 MIB.
AUGMENTS	In some cases, it is useful to add a column to an existing table. The AUGMENTS clause allows you to extend a table by adding one or more columns, represented by some other object. This clause requires the name of the table the object will augment.

SMIPv2 defines a new trap type called NOTIFICATION-TYPE, which we will discuss in "[SNMP Notification](#)" later in this chapter. SMIPv2 also introduces new textual conventions that allow managed objects to be created in more abstract ways. RFC 2579 defines the textual conventions used by SNMPv2, which are listed in [Table 2-4](#).

Table 2-4. Textual conventions for SMIPv2

Textual convention	Description
DisplayString	A string of NVT ASCII characters. A DisplayString can be no more than 255 characters in length.
PhysAddress	A media- or physical-level address, represented as an OCTET STRING.
MacAddress	Defines the media-access address for IEEE 802 (the standard for LANs) in canonical order. (In everyday language, this means the Ethernet address.) This address is represented as six octets.
TruthValue	Defines both true and false Boolean values.
TestAndIncr	Used to keep two management stations from modifying the same managed object at the same time.
AutonomousType	An OID used to define a subtree with additional MIB-related definitions.

A pointer to a particular object instance, such as *ifDescr* for interface 3. In this case, the *VariablePointer* would be the OID *ifDescr.3*.

VariablePointer

A pointer to a row in a table. For example, *ifIndex.3* points to the third row in *ifTable*.

RowPointer

Used to manage the creation and deletion of rows in a table, since SNMP has no way of doing this via the protocol itself. *RowStatus* can keep track of the state of a row in a table as well as receive commands for creation and deletion of rows. This textual convention is designed to promote table integrity when more than one manager is updating rows. The following enumerated types define the commands and state variables: *active(1)*, *notInService(2)*, *notReady(3)*, *createAndGo(4)*, *createAndWait(5)*, and *anddestroy(6)*.

RowStatus

Measures the amount of time elapsed between the device's system uptime and some event or occurrence.

TimeStamp

Measures a period of time in hundredths of a second. *TimeInterval* can take any integer value from 0-2147483647.

TimeInterval

An OCTET STRING used to represent date and time information.

DateAndTime

Defines the type of memory an agent uses. The possible values are *other(1)*, *volatile(2)*, *nonVolatile(3)*, *permanent(4)*, and *readOnly(5)*.

StorageType

Denotes a kind of transport service.

TDomain

Denotes the transport service address. *TAddress* is defined to be from 1-255 octets in length.

TAddress

^[2] Canonical order means that the address should be represented with the least-significant bit first.