CHAPTER 1

Introduction

Network management takes place between two major types of systems: those in control, called managing systems, and those observed and controlled, called managed systems. The most common managing system is called a network management system (NMS). Managed systems can include hosts, servers, or network components such as routers or intelligent repeaters.

To promote interoperability, cooperating systems must adhere to a common framework and a common language, called a protocol. In the Internet network management framework, that protocol is the Simple Network Management Protocol (SNMP). SNMP is an application-layer protocol designed to facilitate the exchange of management information between network devices. The SNMP system consists of three parts:

- SNMP manager
- SNMP agent
- MIB

The Internet network management framework is based on the idea of a managing system interfacing to a managed system. The managing system is called a manager, and runs a network management application. The managed system is running an agent that answers requests from the manager. The manager and the managed system converse using the Simple Network Management Protocol (SNMP).

The “conversation” between the manager and the managed system is about the Management Information Base (MIB) that defines all the information that can be seen or changed by the manager. Portions of the MIB may be standard or proprietary and a similar concept of the MIB must be shared by both manager and agent.

SNMP and its MIBs are defined in a combination of system specific language and Abstract Syntax Notation 1 (ASN.1) ASN.1 is a rich definition language, but SNMP uses only a subset, defined in SNMP's Structure of Management Information (SMI). For transmission, SNMP is encoded according to ASN.1's basic encoding rules (BER).

SNMP may be carried over a wide choice of transport protocols. The most common combination is the User Datagram Protocol over the Internet Protocol, UDP/IP. Other possibilities include AppleTalk, Netware, or Ethernet.

SNMP has facilities for identifying the requester and the operational context in which a request is to be performed by the agent, such as read-only or read-write, a MIB subset for a particular group of users, or a subset that may be elsewhere or obtained through other mechanisms (proxy). These facilities are the ones concerned with security.

SNMP has a small number of MIB management operations it can perform for observation and control of MIB information, comprising various ways of reading (get operations), and one way of modifying (set operation).

This chapter includes the following sections:

Management Information Base Overview

In a managed device, specialized low-impact software modules, called agents, access information about the device and make it available to the network management system (NMS). Managed devices maintain values for a number of variables and report those, as required, to the NMS. For example, an agent might report such data as the number of bytes and packets in and out of the device, or the number of broadcast messages sent and received. In the Internet network management framework, each variable is referred to as a managed object, which is anything that an agent can access and report back to the NMS.

All managed objects are contained in the Management Information Base (MIB) database. The managed objects, or variables, can be set or read to provide information on network devices and interfaces. An NMS can control a managed device by sending a message to an agent of that managed device requiring the device to change the value of one or more of its variables.

MIB Source

MIBs may come from various sources:

- **Standard**– on the IETF standards track at Proposed, Draft, or full standard. A Proposed Standard can change somewhat due to implementation experience. A Draft Standard changes somewhat less, with more attention to backward compatibility. A full Internet Standard doesn't change much. At all levels these are published as Requests for Comment (RFCs).
- **Internet Draft**–IETF work in progress. Sometimes the best way to instrument technology is with an Internet Draft MIB, which is typically being worked on by an IETF working group. Such MIBs are somewhat unstable, so it is necessary to capture the specific Internet Draft and to place the MIB within the Cisco Enterprise MIB space (not in the Experimental branch).
- **Cisco**–Cisco enterprise-specific (also called proprietary or private, even though publicly documented). Such MIBs add instrumentation not covered by standard MIBs. As of IOS Version 10.2, Cisco has old MIBs and new MIBs. The old MIBs are from older software versions and often have somewhat unconventional features.
- **Other companies**– non-Cisco enterprise-specific. It is occastionally appropriate to implement a MIB defined by some other company, especially when implemented technology they originated and instrumented. This has similar problems to Internet Drafts in that a version of the MIB definition must be captured, but the MIB itself should remain whereever in the MIB space the originating company placed it so as to easily support existing applications.

MIB Objects

A MIB is a tree where the leaves are individual items of data called objects. An object may be, for example, a counter or a protocol status. MIB objects are also sometimes called variables. Note that the SNMP framework uses object in a somewhat different way than does OSI management. An OSI object

is a network entity, such as a router or a protocol, which has attributes. These OSI attributes and SNMP objects are essentially the same concept, that is, individual data values. See Appendix A for a detailed description. MIB object consists of the following values:

- Object type–identifies the type of MIB object.
- Syntax–identifies the data type which models the object.
- Access–identifies the maximum level of access and can have one of five values (listed from highest to lowest):
  - Read-create–indicates that instances of the object may be read, written, and created
  - Read-write–indicates that instance of the object may be read or written, but not created
  - Read-only–indicates that instances of the object may be read but not written or created
  - Accessible-for-notify–indicates that instances of the object may only appear in notifications
  - Not-accessible–inidicates that instances of the object may not be directly read, written, or created.
- Status–the status of a managed object can be:
  - Mandatory–indicates that the definition is required and should be implemented
  - Current–indicates that the definition is current
  - Deprecated–indicates that the definition will soon be made obsolete and need no longer be implemented
  - Obsolete–indicates that managed nodes should not implement the object.
- Description–provides a textual description of the managed object

An example of a MIB object is shown below.

tpTDMIfCollectTimeInterval OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object shows measurement time interval seconds.

::= {tpTDMIfStatTableEntry 1}

For descriptions of supported MIBs and how to use MIBs, see the Cisco MIB web site on CCO at http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml.

MIB Archive

Cisco Management Information Bases (MIBs) are archived in Cisco's FTP server and are accessible via anonymous FTP at the following location: ftp://ftp.cisco.com/pub/mibs

Simple Network Management Protocol

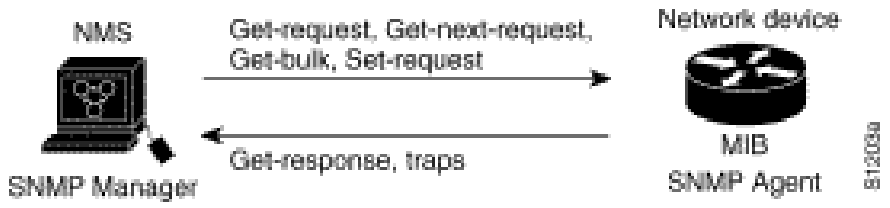
Cisco MIB variables are accessible through the Simple Network Management Protocol (SNMP), which is an application-layer protocol designed to facilitate the exchange of management information between network devices.

Instead of defining a large set of commands, SNMP places all operations in a get-request, get-next-request, and set-request format. For example, an SNMP manager can get a value from an SNMP agent or store a value into that SNMP agent. The SNMP manager can be part of a network management system (NMS), and the SNMP agent can reside on a networking device such as a router. You can compile the Cisco MIB with your network management software. If SNMP is configured on a Catalyst Switch, the SNMP agent can respond to MIB-related queries being sent by the NMS.

An example of an NMS is the CiscoWorks network management software. CiscoWorks uses the Cisco MIB variables to set device variables and to poll devices on the internetwork for specific information. The results of a poll can be displayed as a graph and analyzed in order to troubleshoot internetwork problems, increase network performance, verify the configuration of devices, monitor traffic loads, and so on.

As shown in Figure 1, the SNMP agent gathers data from the MIB, which is the repository for information about device parameters and network data. The agent can send traps, or notification of certain events, to the manager. The Cisco trap file, mib.traps, which documents the format of the Cisco traps, is available on the Cisco host ftp.cisco.com.

Figure 1-1 Simple Network Management Protocol Network



The SNMP manager uses information in the MIB to perform the operations described in Table 1.

Table 1-1 SNMP Manager Operations

Operation	Description
get-request	Retrieve a value from a specific variable.
get-next-request	Retrieve a value from a variable within a table1.
get-response	The reply to a get-request, get-next-request, and set-request sent by an NMS.
set-request	Store a value in a specific variable.
trap	An unsolicited message sent by an SNMP agent to an SNMP manager indicating that some event has occurred.

With this operation, an SNMP manager does not need to know the exact variable name. A sequential search is performed to find the needed variable from within a table.

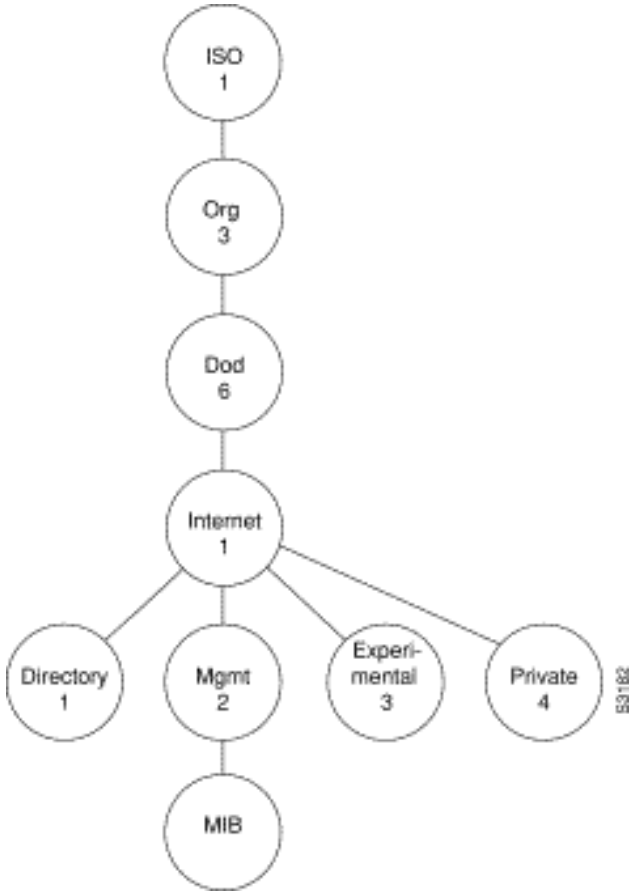
Internet MIB Hierarchy

The MIB structure is logically represented by a tree hierarchy (see Figure 1-2). The structure uses branches and those that fall below each category have short text strings and integers to identify them. Text strings describe object names, while integers allow computer software to create compact, encoded representations of the names. For example, the Cisco MIB variable authAddr is an object name and is denoted by number 5, which is listed at the end of its object identifier number 1.3.6.1.4.1.9.2.1.5.

The object identifier in the Internet MIB hierarchy is the sequence of numeric labels on the nodes along a path from the root to the object. The Internet standard MIB is represented by the object identifier 1.3.6.1.2.1. It also can be expressed as iso.org.dod.internet.mgmt.mib. (See Figure 1-2.)

Note The International Telecommunications Union Telecommunication Standardization Sector (ITU-T) carries out the functions of the former CCITT.

Figure 1-2 Internet MIB Hierarchy



SNMP MIB

An SNMP MIB is an abstract database, that is, a conceptual specification for information that a management application may read and modify in a certain form. This does not imply that the information is kept in the managed system in that same form. The SNMP agent translates between the internal data structures and formats of the managed system and the external data structures and formats defined for the MIB.

The SNMP MIB is conceptually a tree structure with conceptual tables, as discussed in more detail below. Relative to this tree structure, the term "MIB" is used in two senses. In one sense it is actually a MIB branch, usually containing information for a single aspect of technology, such as a transmission medium or a routing protocol.

A MIB used in this sense is more accurately called a MIB module, and is usually defined in a single document. In the other sense a MIB is a collection of such branches.

Such a collection might comprise, for example, all the MIB modules implemented by a given agent, or the entire collection of MIB modules defined for SNMP.

MIBs may be standard or enterprise. Internet standard MIBs are defined by working groups of the Internet Engineering Task Force (IETF) and published as Requests for Comment (RFCs). enterprise MIBs are defined by other organizations, usually individual companies. Done properly, enterprise MIBs instrument technology not covered by standard MIBs, whether completely or as an extension to a standard MIB.

The prototypical standard MIB is MIB-II, the second revision of the original SNMP MIB. MIB-II contained branches for the basic areas of instrumentation, such as the system, its network interfaces, IP, and TCP. All of these started out in a single MIB module, but as SNMPv2 evolves, they are being split into separate modules.

Compliance

Cisco MIBs are a set of variables that are private extensions to the Internet standard MIB II. The MIB II is documented in RFC 1213, Management Information Base for Network Management of TCP/IP-based Internets: MIB-I. It includes information on the benefits of the new feature, supported platforms, related documents, troubleshooting tips, configuration examples, and a detailed command reference.

Cisco Compliance

At present, Cisco implementations of standard MIBs are often read-only or have some objects or object groups missing due to security concerns and time pressure for implementation. Since IOS Version 10.2, the developer must document such specifics with AGENT-CAPABILITIES from RFC 1904.

Implementation

To find what MIBs Cisco implements, start at ftp-eng.cisco.com with ftp://ftp-eng.cisco.com/pub/mibs/README.

This supplies a list of what MIBs are available in what software versions. This can not account for MIBs not included in a particular software subset or because a feature is turned off. This is the function of AGENT-CAPABILITIES descriptions and the snmpORTable (RFC 1907) in later software versions.

The documentation for how to implement a MIB is somewhat sketchy and distributed. As pieces of it coalesce, pointers will appear here. There are three major steps to implementing a MIB:

1. Design the MIB.
2. Generate skeleton code.
3. Add your code.

Once your MIB description document is complete, you run it through a MIB compiler to generate skeleton code to which you will add your own code. Our agent already exists, as do strict, detailed procedures for generating skeleton code. How your code fits into that structure depends somewhat on the access you have to the actual instrumentation and Cisco's conventions for modularity.

The primary documentation for this process is in the form of implemented MIBs and the section on establishing a MIB in the Writing MIBs appendix of the IOS

The following documentation is also available:

SNMP MIB Tables

Tables are a powerful and often confusing aspect of SNMP MIBs. Architectural purists say SNMP has conceptual tables, not real tables. This is because every object, whether in a table or not, is a leaf of the tree, identified by an OID that includes an instance. So, in an abstract sense, all objects are alike. But practically speaking, SNMP has tables and using or implementing them gets somewhat more complex than implementing scalars.

Tables have a rigid structure, defined in the SMI. Tables may contain only simple objects, not other tables, although multiple indexes can represent the concept of tables in tables. An entry, or row, in a table is uniquely identified by one or more table indexes, also called auxilliary objects. The OID of an object from a table is the OID for that object's position in the MIB tree concatenated with a representation of all the table indexes for an entry in the table.

For example, the Interface MIB (RFC 1573) has a key table called the ifTable. Its index object is ifIndex, an integer. Minus the instance, the OID for a counter from that table is:

iso.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifInOctets

Or, numerically:

1.3.6.1.2.1.2.2.1.10

For the interface with ifIndex 7, the full OID is:

iso.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifInOctets.7

1.3.6.1.2.1.2.2.1.10.7

Observe that row selection (instance) comes after column selection. This can be particularly confusing when applying the principle of lexical order to a table. Using the GetNext protocol operation to walk a table proceeds columnwise, that is, all instances for a column are returned before starting the next column.

Table indexes can be much more complex. Here's an example from the Cisco VINES MIB. The INDEX clause from the ASN.1 definition is:

```
INDEX { cvForwNeighborHost,
        ifIndex,
        cvForwNeighborPhysAddress }
```

The first two indexes are simple integers, with ifIndex being imported from the standard ifTable. The final index is a variable length octet string. Including the integers is simple and obvious. The variable-length index object gets a bit more complex. RFC 1212 includes rules for encoding variable length index objects as instances. The general rule is that the value is preceeded by a length and the length and each part of the value are separate subidentifiers.

So, for example, if we have neighbor host number 9, ifIndex 3, and an Ethernet neighbor physical address 0000.0c03.1ef0, the instance portion of an object for that row is: 9.3.6.0.0.12.3.30.240

In RFC 1902, SNMPv2 extends the instance encoding rules to include an "IMPLIED" keyward that can be used on the final instance object if it is variable length. When "IMPLIED" is present, that part of the instance does not have a length in front of it.

Note that lexical ordering for variable length instance objects effectively sorts them by length, so your ASCII text index won't come out naturally in alphabetical order.