

Introduction to Cisco MIBs

This online reference provides a basic description about Management Information Base (MIB) objects and how they are used with Enterprise switches.

This section contains the following information:

- MIB Definition
- Cisco MIB Files
- SNMP Description
- MIB Hierarchy

MIB Definition

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). (Refer to “SNMP Description” for additional information.)

In a managed device, specialized low-impact software modules, called *agents*, access information about the device and make it available to the 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,

MIB Definition

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 MIB, which is a database of the managed objects. 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.

Object Identifier

Each MIB variable is assigned an object identifier. The *object identifier* is the sequence of numeric labels on the nodes along a path from the root to the object. For example, the MIB variable *tftpHost* is indicated by the number 1. The object identifier for *tftpHost* is *iso.org.dod.internet.private.enterprise.cisco.workgroup.products.stack.group.tftp.group.tftpHost variable* or *1.3.6.1.4.1.9.5.1.5.1*. Figure 1-1 provides a visual representation of this variable. The last value is the number of the MIB variable *tftpHost*. Refer to the section “MIB Hierarchy” for additional information.

Figure 1-1 Object Identifier Example for a MIB Variable

iso.org.dod.internet.private.enterprise.cisco
workgroup.products.stack.group.tftp
group.[MIB Variable]
1.3.6.1.4.1.9.5.1.5.[MIB Variable]
└─── *tftpHost [1]*

55899

Tables

When network management protocols use names of MIB variables in messages, each name has an appended suffix. This suffix is called an *instance identifier*. For simple variables, the instance identifier 0 refers to the instance of the variable with that name. A MIB also can contain tables of related variables.

An excerpt of the information on the local IP routing table (known as *lipRoutingTable*) from the associated MIB file follows:

Note The local IP routing table is not supported by Catalyst 5000 series switches and is used as an example only.

```
lipRoutingTable OBJECT-TYPE
    SYNTAX SEQUENCE OF lipRouteEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "A list of IP routing entries."
    ::= { lip 2 }

lipRouteEntry OBJECT-TYPE
    SYNTAX lipRouteEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "A collection of additional objects in the
        cisco IP routing implementation."
    INDEX { ipRouteDest }
    ::= { lipRoutingTable 1 }

lipRouteEntry ::=
    SEQUENCE {
        locRtMask
        lIpAddress,
        locRtCount
        INTEGER,
    }
```

In the example, the *lipRoutingTable* contains two variables: *locRtMask* and *locRtCount*. The index for this table is the destination address of the IP route, or *ipRouteDest*. With *n* number of routes available to a device, *n* rows are available in the IP routing table.

Typically, an instance identifier might be a unique interface number or a 0. An instance identifier can also be an IP address. For example, to find the network mask for the route with a destination address of *131.104.211.243*, use the variable *locRtMask* (locate route mask) with an instance identifier of *131.104.211.243*. The format is *locRtMask.131.104.211.243*.

For additional information on instance identifiers, refer to the section “Defining Instance Identifiers” and “Displaying the IF-INDEX.”

Cisco MIB Files

The Cisco MIB for Cisco IOS is provided with all Cisco software releases and with CiscoWorks router management software.

The Cisco MIB is 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-II*.

Many other Internet-standard MIBs are supported by Cisco agents. These standard MIBs are defined in publications called *Requests for Comments* (RFCs). To find specific MIB information, you must examine the Cisco proprietary MIB structure and the standard RFC MIBs supported by Cisco.

If your NMS is unable to get requested information from a managed device such as a Cisco router, the MIB that allows that specific data collection might be missing. Typically, if an NMS cannot retrieve a particular MIB variable, either the NMS does not recognize the MIB variable, or the agent does not support the MIB variable. If the NMS does not recognize a specified MIB variable, you might need to load the MIB into the NMS, usually with a MIB compiler. For example, you might need to load the Cisco Workgroup MIB or the supported RFC MIB into the NMS to execute a specified data collection. If the agent does not support a specified MIB variable, you must find out what version of Cisco IOS or system software you are running. Different MIBs are supported in different software releases.

Cisco MIB File Subdirectories

Cisco's public MIB files are organized into two subdirectories: SNMPv1 MIBs are in the `v1/` subdirectory and SNMPv2 MIBs are in the `v2/` subdirectory. Determine which MIBs the switch supports and get a description of the files by retrieving the following:

- `supportlists.txt` file—Describes which MIBs the switch supports
- `v1/README`—Describes the SNMPv1 MIBs in the subdirectory
- `v2/README`—Describes the SNMPv2 MIBs in the subdirectory

MIB-Loading Order

Many MIBs use definitions that are defined in other MIBs. These definitions are listed in the section titled `IMPORTS` near the top of the MIB.

If MIB B imports a definition from MIB A, some MIB compilers require you to load MIB A prior to loading MIB B. If you get the MIB loading order wrong, the MIB compiler might complain about what was imported claiming it as *undefined* or not listed in `IMPORTS`. If this happens, look at the loading order of the MIB definitions from the `IMPORTS` of the MIB. Make sure that you have loaded all the preceding MIBs first.

Following is a list of MIBs from which many other MIBs import definitions and the order in which you should load these MIBs:

- 1 `SNMPv2-SMI.my`
- 2 `SNMPv2-TC.my`
- 3 `SNMPv2-MIB.my`
- 4 `RFC1213-MIB.my`
- 5 `IF-MIB.my`
- 6 `CISCO-SMI.my`
- 7 `CISCO-PRODUCTS-MIB.my`
- 8 `CISCO-TC.my`

If you load the MIBs in this order, you can eliminate 95 percent of your load-order definition problems.

SNMP Description

Note You can load most other MIBs (not in the list shown in the section “MIB-Loading Order”) in any order.

SNMP Description

You can access the Cisco MIB variables through the SNMP, 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, and MIB.

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 switch. You can compile the Cisco MIB with your network management software. If SNMP is configured on a Catalyst 5000 series switch, the SNMP agent responds to MIB-related queries being sent by the NMS.

As shown in Figure 1-2, the SNMP agent gathers data from the MIB. Then, the agent can send traps, or notifications of certain events, to the manager. The SNMP manager uses information in the MIB to perform the operations described in Table 1-1.

Figure 1-2 **SNMP Network**

Table 1-1 SNMP Manager Operations

Operation	Description
get-request	Retrieve a value from a specific variable.
get-next-request	Retrieve the value following the named variable. Often used to retrieve variables from within a table. ¹
get-bulk ²	Retrieve large blocks of data, such as multiple rows in a table, which would otherwise require the transmission of many small blocks of data.
set-request	Store a value in a specific variable.
get-response	The reply to the above commands sent by an NMS.
trap	An unsolicited message sent by an SNMP agent to an SNMP manager indicating that some event has occurred.

1. 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 the MIB.

2. The **get-bulk** command only works with SNMP version 2.

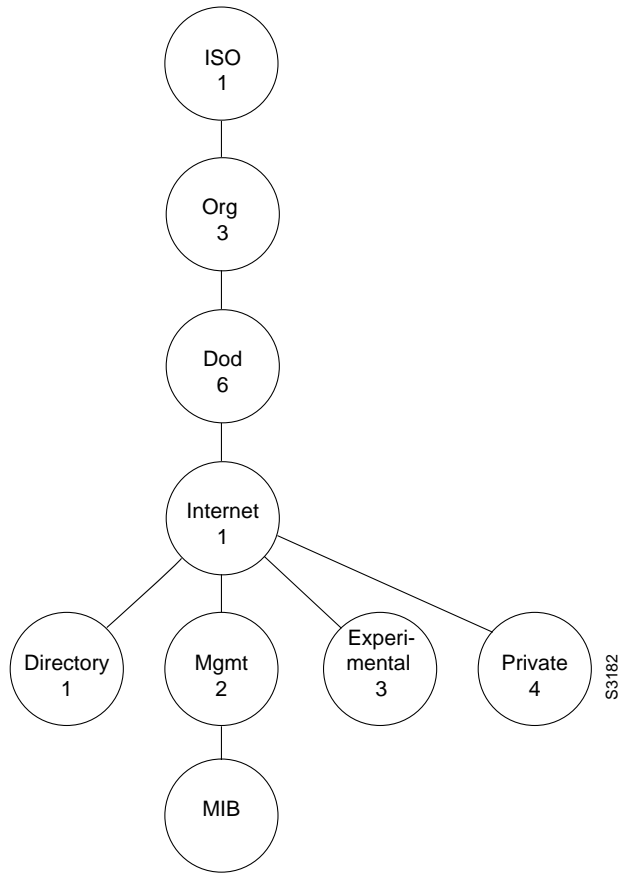
MIB Hierarchy

The MIB structure is logically represented by a tree hierarchy. (See Figure 1-3.) The *root* of the tree is unnamed and splits into three main branches: Consultative Committee for International Telegraph and Telephone (CCITT), International Organization for Standardization (ISO), and joint ISO/CCITT.

These 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 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 1-3.)

Figure 1-3 Internet MIB Hierarchy



Cisco MIB Hierarchy

The object identifier *1.3.6.1.4.1.9*, or *iso.org.dod.internet.private.enterprise.cisco* represents the Cisco Workgroup MIB. The Cisco Workgroup MIB splits into two main areas: Workgroup Products and Cisco Management.

Note Local variables do not change; they are supported by the Cisco Workgroup or Management MIBs in each subsequent system software release. As indicated by the name, temporary variables might have a limited time span or change with each system software release. (See Figure 1-4.)

Catalyst 5000 MIB Hierarchy

In Figure 1-4, the Stack MIB group is identified by 1; its subgroup, called *tftp grp*, is identified by 5. Therefore, a variable in the subgroup *tftp grp* has an object identifier (OID) of *1.3.6.1.4.1.9.5.1.5.n*, where *n* equals the variable number. An appended 0 (for example, *1.3.6.1.4.1.9.5.1.5.n.0*) indicates that there is only one instance of this variable.

LightStream 1010 MIB Hierarchy

In Figure 1-5, the Local MIB group is identified by 2; its subgroups, called *SYSTEM*, are identified by 1 and its subgroup called *ciscoPing*, are identified by 7. Therefore, the variable in the subgroup *ciscoPing* has an OID of *1.3.6.1.4.1.9.2.1.7.0*. The appended 0 indicates that *1.3.6.1.4.1.9.2.1.7.0* is the only instance of this variable.

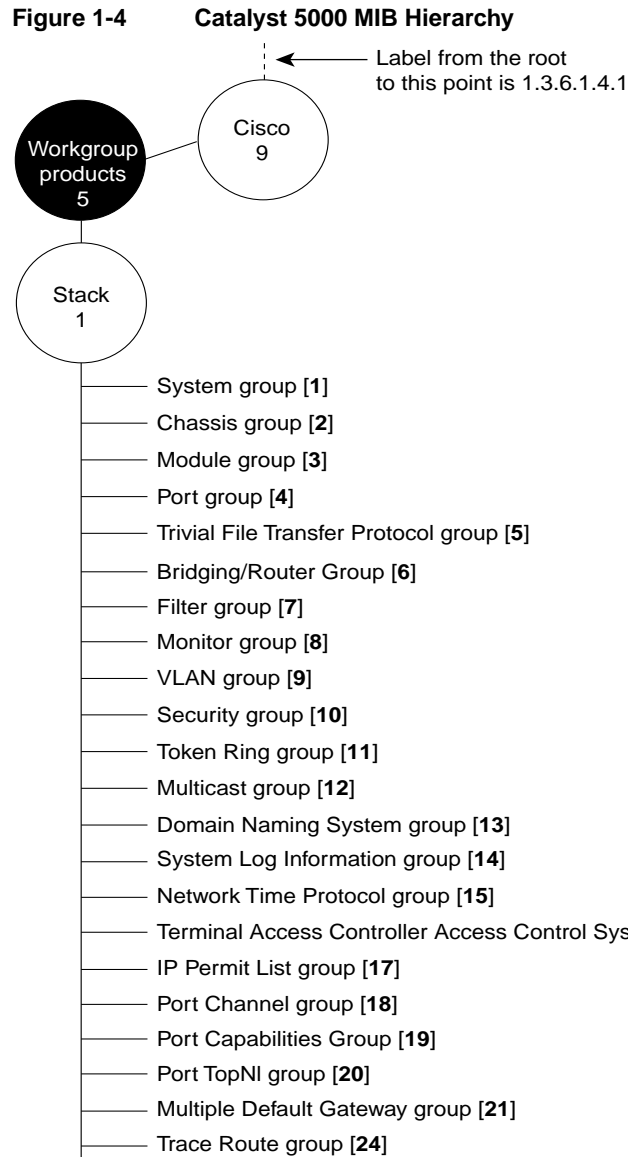
Note Local variables do not change; they are supported by the Cisco MIB in each subsequent system software release. As indicated by the name, temporary variables might have a limited time span or change with each system software release.

Catalyst 3000 MIB Hierarchy

In Figure 1-6, the esStack MIB is identified by 12: its subgroup, called *configuration objects*, is identified by 1. Therefore, a variable in the subgroup *configuration objects* has an OID of *1.3.6.1.4.1.9.5.14.2.n*, where *n* equals the variable number. An appended 0 (for example, *1.3.6.1.4.1.9.5.14.2.n.0*) indicates that there is only one instance of this variable.

Catalyst 8510 MIB Hierarchy

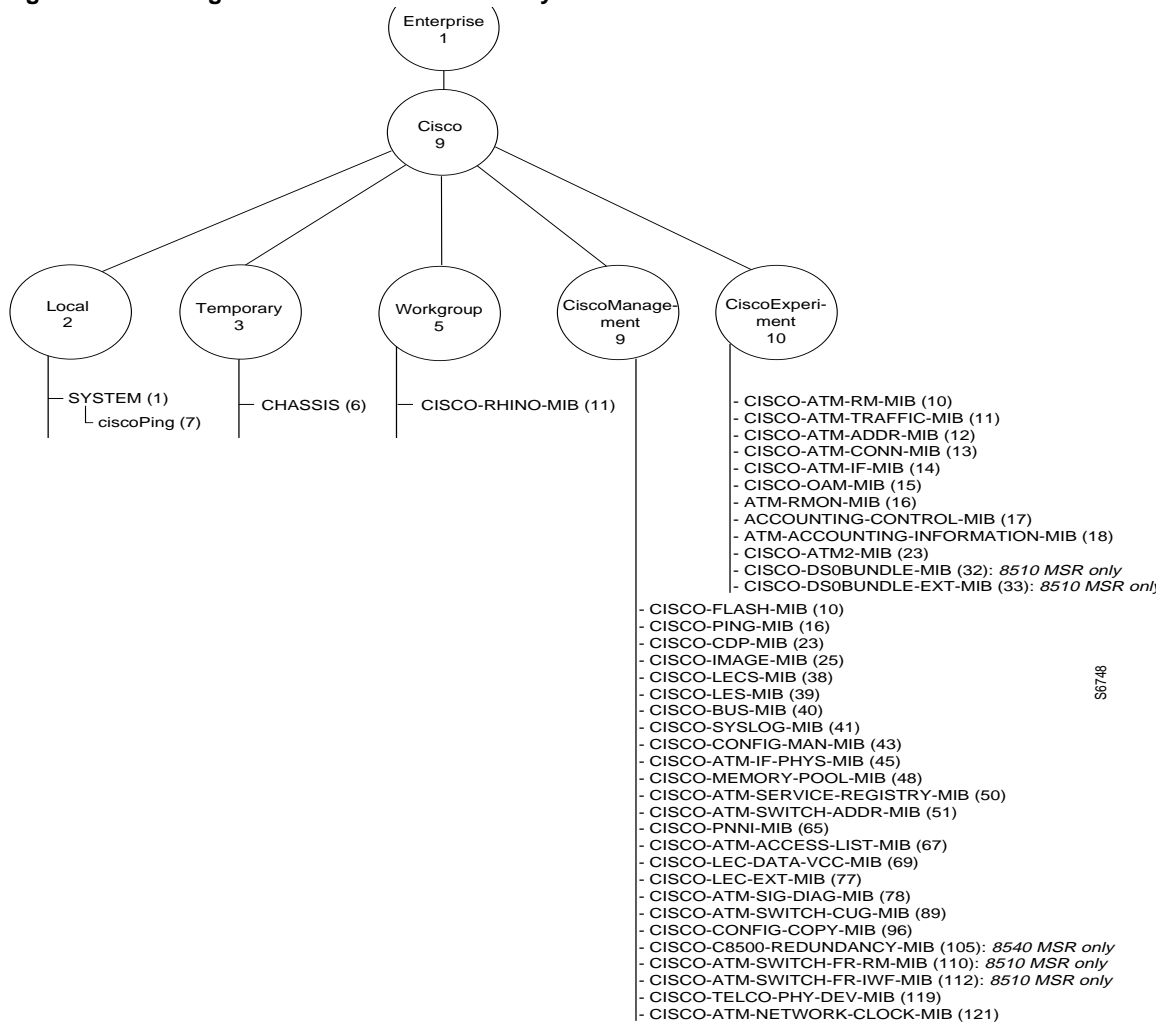
The MIB hierarchy for the Catalyst 8510 are the same as those illustrated for the LightStream 1010 (see Figure 1-5).



S5929

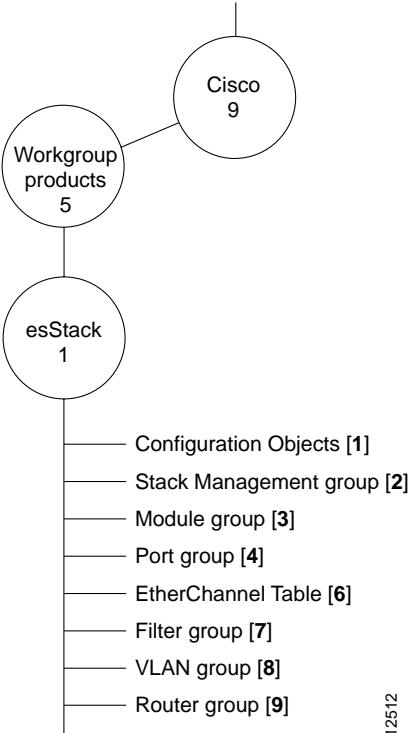
MIB Hierarchy

Figure 1-5 LightStream 1010 MIB Hierarchy



56746

Figure 1-6 Catalyst 3000 MIB Hierarchy



12512

MIB Hierarchy
