

[Send document comments to mxe-doc@cisco.com.](mailto:mxe-doc@cisco.com)



Cisco MXE-OS MIB Quick Reference

Release Date: July 8, 2010

This document describes the private, or local, Simple Network Management Protocol (SNMP) MIB files for the Cisco MXE-OS software.

- [MIBs and Network Management, page 1](#)
- [MIB Guidelines and Limitations for Cisco MXE-OS, page 2](#)
- [Accessing MIB Variables Through SNMP, page 2](#)
- [SNMP Traps and Informs, page 3](#)
- [Interpreting the MIB Structure, page 4](#)
- [About Cisco MIB Files, page 6](#)
- [Accessing and Downloading Cisco MIB Files, page 8](#)
- [Understanding the ENTITY-MIB and Extensions, page 9](#)
- [Extending the IF-MIB, page 9](#)
- [Cisco MXE-OS MIBs for the Cisco MXE 5000 Series, page 10](#)
- [Obtaining Documentation and Submitting a Service Request, page 10](#)
- [Related Documentation, page 11](#)

MIBs and Network Management

The MIB list includes Cisco proprietary MIBs and many other Internet Engineering Task Force (IETF) standard MIBs. These standard MIBs are defined in *Requests for Comments* (RFCs). To find specific MIB information, you must examine the Cisco proprietary MIB structure and related IETF-standard MIBs supported by the Cisco MXE-OS.

Network management takes place between two major types of systems: those systems in control, called *managing systems*, and those systems that managing systems observe and control, 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 switches and routers.



Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Send document comments to mxe-doc@cisco.com.

To promote interoperability, cooperating systems must adhere to a common framework and a common language, called a *protocol*. In the Internet-standard management framework, that protocol is the Simple Network Management Protocol (SNMP).

The exchange of information between managed network devices and a robust NMS is essential for reliable performance of a managed network. Because some devices have a limited ability to run management software, most of the computer processing burden is assumed by the NMS. The NMS runs the network management applications, such as Data Center Network Manager, that present management information to network managers and other users.

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 sent or received by the device or the number of broadcast messages sent and received. In SNMP, each of these variables is referred to as a *managed object*. A managed object is anything that can be managed or 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.

An NMS can control a managed device by sending a request to an agent of that managed device, requiring the device to change the value of one or more of its variables. The managed devices can respond to requests such as **set** or **get**. The NMS uses the **set** request to control the device. The NMS uses the **get** requests to monitor the device. The **set** and **get** requests are synchronous events, which means that the NMS initiates the activity, and the SNMP agent responds.

The managed device can send asynchronous events, or SNMP notifications, to the NMS to inform the NMS of some recent event. SNMP notifications (traps or informs) which are included in many MIBs and allow the NMS to less frequently send get requests to the managed devices.

MIB Guidelines and Limitations for Cisco MXE-OS

The following guidelines and limitations apply to Cisco MXE-OS MIBs:

- Cisco MXE-OS does not support MIB Set operations.
- Cisco MXE-OS supports only SNMP Get.
- By default, all supported traps are enabled in Cisco MXE-OS.
- Cisco MXE-OS does not support informs.
- Cisco MXE-OS does not support SNMPv3 traps.

Accessing MIB Variables Through SNMP

The SNMP system consists of three parts: the SNMP manager, the SNMP agent, and the MIB. You can compile Cisco MIBs with your network management software. If SNMP is configured on a device, the SNMP agent responds to MIB-related queries sent by the NMS.

SNMPv1 was the initial version of the protocol. SNMPv2 added support for 64-bit counters, and SNMPv3 added increased security for access, authentication, and encryption of managed data.

[Table 1](#) describes the SNMP operations.

Send document comments to mxe-doc@cisco.com.

Table 1 **SNMP Operations**

Operation	Description
get-request	Retrieves a value from a specific variable.
get-next-request	Retrieves the value following the named variable. Often used to retrieve variables from within a table. ¹
get-bulk ²	Retrieves 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 ³	Stores a value in a specific variable.
response	Replies to the above commands sent by an NMS and to the informs sent by an agent.
trap	Sends an unsolicited message by an SNMP agent to an SNMP manager indicating that some event has occurred.
inform ²	Sends an unsolicited message by an SNMP agent to an SNMP manager indicating that some event has occurred. Differs from a trap in that an acknowledgement is required from the manager.

1. With this operation, an SNMP manager does not need to know the exact variable name. A sequential search finds the next variable from within the MIB.
2. The **get-bulk** and **inform** commands are not a part of SNMPv1.
3. Cisco MXE-OS Release 1.0 does not support the **set-request** command.

SNMP Traps and Informs

You can configure the software to send notifications to SNMP managers when particular events occur. You can send these notifications as traps or inform requests. Traps are unreliable because the receiver does not send any acknowledgment when it receives a trap. The sender cannot determine if the trap was received. However, an SNMP manager that receives an inform request acknowledges the message with an SNMP response. If the sender never receives a response, the inform request can be sent again. Informs are more likely to reach their intended destinations than traps.

Notifications may contain a list of MIB variables (varbinds) that clarify the status this is relayed by the notification. The list of varbinds associated with a notification is included in the notification definition in the MIB. In the case of standard MIBs, Cisco has enhanced some notifications with additional varbinds that further clarify the cause of the notification. See the [“Extending the IF-MIB” section on page 9](#) for an example of these extensions in the IF-MIB.

Use the SNMP-TARGET-MIB to obtain more information on notification destinations. See the [Configuration Guide for Cisco MXE-OS](#).



Note

- By default, all supported traps are enabled in Cisco MXE-OS.
- Cisco MXE-OS does not support informs.

Send document comments to mxe-doc@cisco.com.

Interpreting the MIB Structure

A MIB presents the managed data in a logical tree hierarchy, using an IETF standard syntax called the Structure of Management Information (SMI). Branches of this MIB tree are organized into individual tables, which contain the managed data as leaf objects.

- [Object Identifiers, page 4](#)
- [Tables, page 4](#)
- [SYNTAX Clause, page 5](#)
- [MAX-ACCESS Clause, page 5](#)
- [STATUS Clause, page 6](#)
- [AGENT-CAPABILITIES, page 6](#)

Object Identifiers

The MIB structure is logically represented by a tree hierarchy. 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 branches 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.

Each MIB variable is assigned with 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.enterprises.cisco.workgroup.ciscoStackMIB.tftpGrp.tftpHost or .1.3.6.1.4.1.9.5.1.5.1. The last value is the number of the MIB variable tftpHost.

Tables

SNMP MIBs organize information into tables. When network management protocols use names of MIB objects in messages, each name has an appended suffix. This suffix is called an *instance identifier*. It identifies one occurrence of the associated MIB object. For simple scalar objects, the instance identifier 0 refers to the instance of the object with that name (for example, sysUpTime.0).

A MIB also can contain tables of related objects. For example, ifOperStatus is a MIB object inside the ifTable from the IF-MIB. It reports the operational state for an interface on a device. Because devices may have more than one interface, it is necessary to have more than one instance of ifOperStatus. This instance value is added to the end of the MIB object as the instance identifier (for example, ifOperStatus.2 reports the operational state for interface number 2).

Each object in a table is constructed with a set of clauses defined by the SMI. These clauses include the SYNTAX clause, MAX-ACCESS clause, STATUS clause, and DESCRIPTION clause.

An excerpt of the information in the VSAN table (known as vsanTable) from CISCO-VSAN-MIB follows:

```
vsanTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF VsanEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
```

Send document comments to mxe-doc@cisco.com.

```

        "A table of VSANs configured on this device."
        ::= { vsanConfiguration 3 }
vsanEntry OBJECT-TYPE
    SYNTAX      VsanEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry (conceptual row) in the vsanTable."
    INDEX { vsanIndex }
    ::= { vsanTable 1 }
VsanEntry ::= SEQUENCE {
    vsanIndex          VsanIndex,
    vsanName           SnmpAdminString,
}

```

In the example, vsanTable contains two variables: vsanIndex and vsanName. (There are more values in the actual vsanTable.) The index for this table is the ID of the VSAN, or vsanIndex. With n number of VSANs configured, n rows are present in the table. If you want to retrieve the vsanName that matches VSAN ID 3 (vsanIndex is 3), then you would issue an SNMP **get** for vsanName.3.

SYNTAX Clause

The SYNTAX clause describes the format of the information, or value, that is returned when you monitor or set information in a MIB.

The Cisco MXE-OS MIBs are defined with the SNMPv2 Structure of Management Information version 2 (SNMPv2-SMI) defined in RFC 1902. Some examples of SNMPv2-SMI syntax are as follows:

- Counter32—A nonnegative integer that increases until it reaches some maximum value. After reaching the maximum value, it rolls over to zero. For example, the variable ifInOctets, with a Counter32 syntax, counts the number of input octets on an interface.
- Gauge32—A nonnegative integer that increases until it reaches some maximum value. After reaching the maximum value, it stays fixed (no roll over).
- Counter64—A nonnegative 64-bit integer that increases until it reaches some maximum value. After reaching the maximum value, it rolls back to zero. Counter64 is used for MIB objects that can reach high values in a short period of time (for example, a packet counter for a Gigabit Ethernet port).
- Integer32—An integer from -2^{32} to $2^{32}-1$.
- IPAddress—An octet string that represents an IP address. For example, the variable hostConfigAddr indicates the IP address of the host that provided the host configuration file for a device.
- Timeticks—A nonnegative integer that counts the hundredths of a second that have elapsed since an event. For example, the variable loctcpConnElapsed provides the length of time that a TCP connection has been established.

MAX-ACCESS Clause



Note

- Cisco MXE-OS does not support MIB Set operations.
- Cisco MXE-OS supports only SNMP Get.

Send document comments to mxe-doc@cisco.com.

The MAX-ACCESS clause identifies the maximum access level for the associated MIB object. This clause can represent one of the following five states:

- read-create—You can read, modify, or create objects as rows in a table.
- read-write—You can read or modify this object.
- read-only—You can only read this object.
- accessible-for-notify—You cannot read or write to this object. SNMP notifications can send this object as part of their event information.
- not-accessible—You cannot read or write to this object. Table indices are typically objects that are not accessible.

STATUS Clause

The STATUS clause identifies whether the definition is current or historic. This clause can represent one of the following states:

- current—The definition is current and supported.
- obsolete—The definition is obsolete and not supported.
- deprecated—The definition is obsolete but supported to foster interoperability with older implementations.

When one table or object is deprecated by other, both can appear in the MIB output because deprecated tables or objects are not always removed. The STATUS clause specifies which definition is valid.

AGENT-CAPABILITIES

In SNMP, capabilities files provide implementation details for the associated MIB. These files, called AGENT-CAPABILITIES, list supported conformance groups and any deviations from the MIB as implemented in the associated software version. For example, the CISCO-AAA-SERVER-CAPABILITY provides the implementation details for the CISCO-AAA-SERVER-MIB, as implemented in Cisco MXE-OS.



Note

Capabilities files may have implementation details for more than one software release. You need to match your software release to the corresponding AGENT-CAPABILITIES clause in this file.

About Cisco MIB Files

MIB II is documented in RFC 1213, *Management Information Base for Network Management of TCP/IP-based Internets: MIB-II*. Portions of MIB-II have been updated since RFC 1213. See the IETF website <http://www.ietf.org> for the latest updates to this MIB.

If your NMS cannot get requested information from a managed device, such as a Cisco switch, 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

Send document comments to mxe-doc@cisco.com.

private 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 system software that you are running. Different software releases support different MIBs.

**Note**

Cisco and IETF MIBs are updated frequently. You should download the latest MIBs from Cisco.com whenever you upgrade Cisco MXE-OS.

Cisco MIB File Directories

You can find Cisco MIBs at the following URL:

<http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml>

Cisco MIB files are organized into two directories: SNMPv1-SMI MIBs are in the SNMPv1 directory and SNMPv2-SMI MIBs are in the SNMPv2 directory. Cisco also includes supported IETF-standard MIBs at this website. Use this support list to access and download the individual MIB files.

MIB Loading Order

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

For example, 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, you might get an error message about what was imported, claiming it is undefined or not listed in IMPORTS. If you receive an error, 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.

Here is a list of MIBs from which many other MIBs import definitions. The MIBs are listed in the order in which you should load them:

- SNMPv2-SMI.my
- SNMPv2-TC.my
- SNMPv2-MIB.my
- RFC1213-MIB.my
- IF-MIB.my
- CISCO-SMI.my
- CISCO-TC.my
- CISCO-ST-TC.my
- ENTITY-MIB.my

If you load the MIBs in this order, you can eliminate most of your load-order definition problems. You can load most other MIBs (those not listed here) in any order.

Send document comments to mxe-doc@cisco.com.

Accessing and Downloading Cisco MIB Files

You can use passive FTP to access and download the Cisco MIB files.

- [Enabling Passive FTP on Internet Explorer, page 8](#)
- [Using Passive FTP to Access and Download the MIB Files, page 8](#)



Note

You can also access and download Cisco MIB files using the SNMP Object Navigator tool located at the following site: <http://tools.cisco.com/Support/SNMP/do/BrowseMIB.do?local=en>. You can use this tool to translate SNMP object identifiers (OIDs) into object names, search object names, descriptions, browse OID trees, and download MIB files.

Enabling Passive FTP on Internet Explorer

If you are using Internet Explorer, you might need to enable passive FTP.

Procedure

- Step 1** Open Internet Explorer, and click **Tools > Internet Options**.
- Step 2** Click the **Advanced** tab on the top of the window.
- Step 3** Scroll down to the **Browsing** category.
- Step 4** Check the **Use Passive FTP [for firewall and DSL modem compatibility]** check box.
- Step 5** Click **OK** to save changes.

Using Passive FTP to Access and Download the MIB Files

To access MIB files using passive FTP, you must know the names of the MIB files that you want to download. See the [“Using HTTP to Access and Download the MIB Files from Cisco.com” section on page 8](#) to access the Cisco MXE-OS support list for the names of supported MIBs. These steps assume that your passive FTP utility has UNIX-like commands.

Procedure

- Step 1** Access [ftp.cisco.com](ftp://ftp.cisco.com) using passive FTP.
- Step 2** Log in with your Cisco.com username and password, or as anonymous, with your e-mail address.
- Step 3** Enter **cd /pub/mibs/v2/** to change directories.
- Step 4** Use the **get** command to copy the desired files to your local system.
- Step 5** Use the **quit** command to exit passive FTP.

Send document comments to mxe-doc@cisco.com.

Understanding the ENTITY-MIB and Extensions

The ENTITY-MIB provides basic management and identification of physical and logical entities within a network device. Cisco MXE-OS support for the ENTITY-MIB focuses on the physical entities within a device. This MIB provides details on each module, power supply, and fan tray within a switch chassis. It gives enough information to correctly map the containment of these entities within the switch, building up a chassis view.

Cisco has developed a number of private extensions to the ENTITY-MIB to provide more details for these physical entities. Each MIB extensions shares the common index value, `entPhysicalIndex`, which allows the management application developer to link information across multiple MIBs.

[Table 2](#) lists the Cisco MIB extensions that are linked to the ENTITY-MIB by `entPhysicalIndex`.

Table 2 **ENTITY-MIB Extensions**

MIB	Description
CISCO-ENTITY-ASSET-MIB	Provides manufacturing asset number and revision information per physical entity in the switch.
CISCO-ENTITY-EXT-MIB	Extends the <code>entityPhysicalTable</code> for modules with processors. For each of these modules, this MIB provides memory statistics and LED information.
CISCO-ENTITY-FRU-CONTROL-MIB	Manages field-replaceable units, such as power supplies, fans, and modules.
CISCO-ENTITY-SENSOR-MIB	Provides sensor data for environmental monitors such as temperature gauges.
CISCO-IMAGE-UPGRADE-MIB	Provides module image management based on <code>entPhysicalIndex</code> .

Extending the IF-MIB

The IF-MIB provides basic management status and control of interfaces and sublayers within a network device. Multiple standard and Cisco-specific MIBs use `ifIndex` from the IF-MIB to extend management for specific interface types. Cisco MIBs also enhance the two interface notifications, `linkUp` and `linkDown`, from the IF-MIB to provide a clearer indication of the reason for these notifications. Cisco MIBs add up to two `varbinds` to `linkUp` and `linkDown`, as shown in [Table 3](#).

Table 3 **Varbinds Added to IF-MIB Notifications**

Notification	Varbinds Added
<code>linkUp</code>	<code>ifDescr</code>
<code>linkDown</code>	<code>ifDescr</code>

Send document comments to mxe-doc@cisco.com.

Cisco MXE-OS MIBs for the Cisco MXE 5000 Series

The following is a list of Cisco MXE-OS MIBs supported by the Cisco MXE 5000 Series:

- CISCO-ENTITY-ASSET-MIB.my
- CISCO-ENTITY-FRU-CONTROL-MIB.my
- CISCO-ENTITY-SENSOR-MIB.my
- CISCO-ENTITY-VENDORTYPE-OID-MIB.my
- CISCO-NTP-MIB.my
- CISCO-PRODUCTS-MIB.my
- CISCO-SMI.my
- ENTITY-MIB.my
- IANAifType-MIB.my
- IANA-RTPROTO-MIB.my
- IF-MIB.my
- INET-ADDRESS-MIB.my
- IP-FORWARD-MIB.my
- IP-MIB.my
- SNMP-FRAMEWORK-MIB.my
- SNMP-MPD-MIB.my
- SNMP-NOTIFICATION-MIB.my
- SNMP-TARGET-MIB.my
- SNMP-USM-MIB.my
- SNMPv2-CONF.my
- SNMPv2-MIB.my
- SNMPv2-SMI.my
- SNMPv2-TC.my
- SNMP-VACM-MIB.my
- TCP-MIB.my
- UDP-MIB.my

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

<http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html>

Subscribe to the *What's New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS Version 2.0.

Send document comments to mxe-doc@cisco.com.

Related Documentation

You can access Cisco MXE-OS and Cisco Media Experience Engine 5000 Series documentation at the following URL:

http://www.cisco.com/en/US/products/ps10588/tsd_products_support_series_home.html

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2010 Cisco Systems, Inc. All rights reserved.

Send document comments to mxe-doc@cisco.com.