



Cisco Nexus 7000 Series and 9000 Series NX-OS MIB Quick Reference

Preface	ii
Audience	ii
Document Conventions	ii
Documentation Feedback	iii
Communications, Services, and Additional Information	iii
Cisco Nexus 7000 Series and 9000 Series NX-OS MIB Quick Reference	5
MIBs and Network Management	5
Accessing MIB Variables Through SNMP	5
SNMP Traps and Informs	6
Interpreting the MIB Structure	6
About Cisco MIB Files	9
Accessing and Downloading Cisco MIB Files	10
Understanding the ENTITY-MIB and Extensions	10
Extending the IF-MIB	11



Preface

Audience

This publication is for network administrators who configure and maintain Cisco Nexus devices.

Document Conventions



Note As part of our constant endeavor to remodel our documents to meet our customers' requirements, we have modified the manner in which we document configuration tasks. As a result of this, you may find a deviation in the style used to describe these tasks, with the newly included sections of the document following the new format.

Command descriptions use the following conventions:

Convention	Description
bold	Bold text indicates the commands and keywords that you enter literally as shown.
<i>Italic</i>	Italic text indicates arguments for which the user supplies the values.
[x]	Square brackets enclose an optional element (keyword or argument).
[x y]	Square brackets enclosing keywords or arguments separated by a vertical bar indicate an optional choice.
{x y}	Braces enclosing keywords or arguments separated by a vertical bar indicate a required choice.
[x {y z}]	Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.
variable	Indicates a variable for which you supply values, in context where italics cannot be used.

Convention	Description
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.

Examples use the following conventions:

Convention	Description
screen font	Terminal sessions and information the switch displays are in screen font.
boldface screen font	Information you must enter is in boldface screen font.
<i>italic screen font</i>	Arguments for which you supply values are in italic screen font.
<>	Nonprinting characters, such as passwords, are in angle brackets.
[]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

This document uses the following conventions:



Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.



Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to: .

We appreciate your feedback.

Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at [Cisco Profile Manager](#).
- To get the business impact you're looking for with the technologies that matter, visit [Cisco Services](#).
- To submit a service request, visit [Cisco Support](#).
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit [Cisco Marketplace](#).
- To obtain general networking, training, and certification titles, visit [Cisco Press](#).
- To find warranty information for a specific product or product family, access [Cisco Warranty Finder](#).

Cisco Bug Search Tool

[Cisco Bug Search Tool](#) (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.

Revised: August 31, 2023,

Cisco Nexus 7000 Series and 9000 Series NX-OS MIB Quick Reference

This document describes the private, or local, SNMP Management Information Base (MIB) files supported by the software.

MIBs and Network Management

The MIB list includes Cisco proprietary MIBs and many other Internet Engineering Task Force (IETF) standard MIBs. The IETF 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 software.

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.

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 Cisco 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 values, 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, allow the NMS to less frequently send get requests to the managed devices.

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.

This table describes SNMP operations.

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	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 ³	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 command is not a part of SNMPv1.

³ The inform command is not a part of SNMPv1.

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, or *varbinds*, that clarify the status that 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.

Use the SNMP-TARGET-MIB to obtain more information on trap destinations and inform requests.



Note You must enable most notifications through the CLI.

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

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.

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 can also 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
        "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, which is referred to as the 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), 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 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 rollover).

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 `loTcpConnElapsed` provides the length of time that a TCP connection has been established.

MAX-ACCESS Clause

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, read-write, read-only, accessible-for-notify, and not-accessible.

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.

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 in different software releases.



Note Capabilities files might 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 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 your software.

Cisco MIB Support Lists

You can find the list of supported MIBs for your platform at the following URLs:

- Nexus 9000 - <https://cisco.github.io/cisco-mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html>
- Nexus 7000 - <https://cisco.github.io/cisco-mibs/supportlists/nexus7000/Nexus7000MIBSupportList.html>

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.

The following 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-MIB.my
- RFC1213-MIB.my
- IF-MIB.my
- CISCO-SMI.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.

Accessing and Downloading Cisco MIB Files

You can access and download the Cisco MIB files from the following site:

<https://github.com/cisco/cisco-mibs/blob/main/v2>

You can also access and download Cisco MIB files using the SNMP Object Navigator tool located at the following site:



Note This site takes you to Cisco's GitHub MIBs repository)

<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, and descriptions, browse OID trees, and download MIB files.

Understanding the ENTITY-MIB and Extensions

The ENTITY-MIB provides basic management and identification of physical and logical entities within a network device. Cisco NX-OS support for the ENTITY-MIB focuses on the physical entities within a device. This MIB provides details about 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 extension shares the common index value, entPhysicalIndex, which allows the management application developer to link information across multiple MIBs.

This table lists the Cisco MIB extensions that are linked to the ENTITY-MIB by entPhysical Index.

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 entityPhysicalTable 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 entPhysicalIndex.

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 the management for specific interface types. Some Cisco products also add `varbinds` to the IF-MIB interface notifications, `linkUp` and `linkDown`, to provide a clearer indication of the reason for these notifications.

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS REFERENCED IN THIS DOCUMENTATION ARE SUBJECT TO CHANGE WITHOUT NOTICE. EXCEPT AS MAY OTHERWISE BE AGREED BY CISCO IN WRITING, ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS DOCUMENTATION ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

The Cisco End User License Agreement and any supplemental license terms govern your use of any Cisco software, including this product documentation, and are located at: <http://www.cisco.com/go/softwareterms>. Cisco product warranty information is available at <http://www.cisco.com/go/warranty>. US Federal Communications Commission Notices are found here <http://www.cisco.com/c/en/us/products/us-fcc-notice.html>.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any products and features described herein as in development or available at a future date remain in varying stages of development and will be offered on a when-and if-available basis. Any such product or feature roadmaps are subject to change at the sole discretion of Cisco and Cisco will have no liability for delay in the delivery or failure to deliver any products or feature roadmap items that may be set forth in this document.

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

The documentation set for this product strives to use bias-free language. For the purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: [www.cisco.com go trademarks](http://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. (1721R)

© Cisco Systems, Inc. All rights reserved.



Americas Headquarters
Cisco Systems, Inc.
San Jose, CA 95134-1706
USA

Asia Pacific Headquarters
CiscoSystems(USA)Pte.Ltd.
Singapore

Europe Headquarters
CiscoSystemsInternationalBV
Amsterdam,TheNetherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.