

Local Variables

This chapter describes the MIB variables within the Cisco product line. Certain groups of variables might or might not be present, depending upon the software options and configuration in the managed device.

Local Variables Available

- Flash Group
 - The variables in this group have been deprecated and replaced with the Cisco Flash group, found in the ciscoMgmt tree.
- Interface Group, consisting of
 - Fast Serial Interface Processor (FSIP)
 - Interface Table
 - Across All Interfaces
 - Address Resolution Protocol (ARP)
 - AppleTalk
 - Apollo
 - Bridging
 - Connectionless Network Service (CLNS)
 - DECnet
 - HP Probe
 - Internet Protocol (IP)

Local Variables Available

- LAN Network Manager (LNM)
- Maintenance Operation Protocol (MOP)
- Novell
- Other Protocols
- Serial Tunnel (STUN)
- Spanning Tree
- Banyan Virtual Integrated Network Service (VINES)
- Xerox Network Systems (XNS)
- Internet Protocol (IP) Group
- IP Accounting Group
- IP Checkpoint Accounting Group
- System Group, consisting of the following. The Host Configuration File variables, which were members of this group, are now obsolete.
 - Basic
 - Buffer
 - Buffer Elements
 - Small Buffers
 - Middle Buffers
 - Big Buffers
 - Large Buffers
 - Huge Buffers
 - CPU Utilization
 - Environmental Monitor Card

The variables in this section have been deprecated and replaced with the variables in the Cisco Environmental Monitor group, found in the ciscoMgmt tree.

- Network Configuration File
- System Configuration
- Terminal Services Group
- Transmission Control Protocol (TCP) Group

The variables in this group have been deprecated and replaced with the Cisco Transmission Control Protocol (ciscoTCP) group in the ciscoMgmt tree.

Flash Group

The variables in this section have been deprecated and are replaced by the variables in the Cisco Flash group found in the ciscoMgmt tree.

The Flash memory card is an add-in card of Flash EPROM (erasable programmable read-only memory) storage onto which system software images can be stored, booted, and rewritten.

Flash File Table

The local Flash File table, *lflashFileDirTable*, contains information on a per file basis and includes the following three variables: *flashDirName*, *flashDirSize*, and *flashDirStatus*. The index to this table is *flashEntries*, or the number of Flash files. If the device has *n* number of Flash files, the table will contain *n* number of rows.

For example, in Table 2-1, the *flash1* file has a directory size of 50 octets, and its status is valid, represented by the integer 1.

Table 2-1 **Flash File Table**

flashEntries	flashDirName	flashDirSize	flashDirStatus
1	flash1	50	1
2	flash2	100	1
3	flash3	200	2

Flash Group

flashDirName

Provides the name associated with a Flash directory entry.

Syntax: Display string

Access: Read-only

flashDirSize

Provides the size (in octets) of a Flash directory entry.

Syntax: Integer

Access: Read-only

flashDirStatus

Indicates the status of the Flash directory entry.

Syntax: Integer (1 = valid, 2 = deleted)

Access: Read-only

End of Table

flashcard

Provides the type of card connected to the Flash card installed in the router. For example, the type of card connected to the Flash card could be either CSC-MS or CSC-MC+.

Syntax: Display string

Access: Read-only

flashController

Provides the type of Flash controller (either CCTL or CCTL2) installed in the router.

Syntax: Display string

Access: Read-only

flashEntries

Provides the number of directory entries, or files, that exist in the Flash memory directory.

Syntax: Integer

Access: Read-only

flashErase

Sets a request to erase Flash memory, freeing up all available memory space. All of the Flash memory is erased out. Individual files cannot be erased from Flash memory.

Syntax: Integer

Access: Write-only

flashEraseStatus

Indicates the status of current or last erasing of Flash memory.

Syntax: Integer

Access: Read-only

flashEraseTime

Indicates the value of sysUpTime the last time the Flash memory was erased.

Syntax: Timeticks

Access: Read-only

Flash Group

flashFree

Provides the amount of available Flash memory in octets.

Syntax: Integer

Access: Read-only

flashSize

Provides the amount of total Flash memory in octets.

Syntax: Integer

Access: Read-only

flashStatus

Indicates the status of the availability of Flash memory.

Syntax: Integer

Access: Read-only

flashToNet

Requests to write the Flash memory to a Trivial File Transfer Protocol (TFTP) server. The value (display string) is the name of the Flash file being sent, or written, to the server. The instance ID is the IP address of the TFTP host.

This copy of the system image can serve as a backup copy and can also be used to verify that the copy in the Flash memory is the same as the original file.

The Flash memory card can be used as a TFTP file server for other routers on the network. This feature allows you to boot a remote router with an image that resides in the Flash server memory.

Syntax: Display string

Access: Write-only

flashToNetStatus

Indicates the status of the current or last flash to net transfer.

Syntax: Integer

Access: Read-only

flashToNetTime

Indicates the value of sysUpTime the last time a file was copied from the Flash memory in the router to the TFTP host.

Syntax: Timeticks

Access: Read-only

flashVPP

Provides the status of the VPP DIP jumper on the Flash memory card. Files can be written to the Flash memory card only if the VPP DIP jumper is turned on.

Syntax: Integer (1 = VPP enabled/Flash write enabled, 2 = VPP disabled/Flash write disabled)

Access: Read-only

netToFlash

Copies a software image from Trivial File Transfer Protocol (TFTP) server to the Flash memory on the router. The value (display string) is the name of the file being sent, or written, to the Flash memory. The instance ID is the IP address of the TFTP host.

The TFTP image copied to the Flash memory must be at least System Software Release 9.0 or later. If earlier system software is copied into the Flash memory, the host processor card will not recognize the CSC-MC+ card upon the next reboot.

Interface Group

If free Flash memory space is unavailable, or if the Flash memory has never been written to, the erase routine is required before new files can be copied.

Syntax: Display string

Access: Write-only

netToFlashStatus

Indicates the status of the current or next-to-last flash transfer.

Syntax: Integer

Access: Read-only

netToFlashTime

Indicates the value of sysUpTime the last time a file was copied from a Trivial File Transfer Protocol (TFTP) server to the Flash memory on the router.

Syntax: Timeticks

Access: Read-only

Interface Group

The following variables apply to interfaces attached to Cisco devices. These variables can be used to monitor the performance of the network in terms of the number of packets dropped, time allocations for input and output packets, and so on. These variables also can be used for fault management. For example, variable values indicate which interfaces are dropping packets or have had to be restarted several times.

Fast Serial Interface Processor (FSIP)

The local FSIP Card table, *fsipTable*, contains information about FSIP cards used by the Cisco 7000 and includes the following six variables that provide information about the processor: *locIfFSIPtype*, *locIfFSIPrts*, *locIfFSIPcts*, *locIfFSIPdtr*, *locIfFSIPdcd*, and *locIfFSIPdsr*. The index to this table is *locIfFSIPIndex*, which indicates the interface index of the card corresponding to its *IfIndex*.

Table 2-2 FSIP Card Table

locIfFSIPIndex	locIfFSIPtype	locIfFSIPrts	locIfFSIPcts	and so on
1	DCE	1	2	
2	DTE	1	3	
and so on				

locIfFSIPcts

Indicates whether the CTS (clear to send) signal is up or down.

Syntax: Integer (1 = not available, 1 = up, 2 = down)

Access: Read-only

locIfFSIPdcd

Indicates whether the DCD (data carrier detect) signal is up or down.

Syntax: Integer (1 = not available, 2 = up, 3 = down)

Access: Read-only

locIfFSIPdsr

Indicates whether the DSR (data set ready) signal is up or down.

Syntax: Integer (1 = not available, 2 = up, 3 = down)

Access: Read-only

Interface Group

locIfFSIPdtr

Indicates whether the DTR (data terminal ready) signal is up or down.

Syntax: Integer (1 = not available, 2 = up, 3 = down)

Access: Read-only

locIfFSIPIndex

Indicates the index interface port of the corresponding *ifIndex*. (RFC 1213)

Syntax: Integer

Access: Read-only

locIfFSIPrts

Indicates whether the RTS (request to send) signal is up or down.

Syntax: Integer (1 = not available, 2 = up, 3 = down)

Access: Read-only

locIfFSIPtype

Indicates whether the FSIP line uses DCE (data communications equipment) or DTE (data terminal equipment).

Syntax: Integer (1 = not available, 2 = DTE, 3 = DCE)

Access: Read-only

Interface Table

The Interface table, *lifTable*, contains all of the variables in the Interface group. The index to the table is *ifIndex*, which indicates the number of the interface. If the device has *n* number of interfaces, the Interface table will contain *n* rows.

In the Interface table shown in Table 2-3, the first column indicates the number of interfaces on the device. Each of the variables in the interface table occupies one column; for example, *locIfHardType* is shown in a column, followed by *locIfLineProt* in the next column, and so on.

Table 2-3 **Interface Table**

Interface Numer	locIfHardType	locIfLineProt	and so on
1	Ethernet	1	
2	TokenRing	0	
3	FDDI	1	
and so on			

Across All Interfaces

This section contains basic interface variables that apply to all interfaces and are not protocol-specific.

locIfCarTrans

Provides the number of times the serial interface received the Carrier Detect (CD) signal. If the carrier detect line is changing state often, it might indicate modem or line problems.

Syntax: Integer

Access: Read-only

locIfCollisions

Provides the number of output collisions detected on this interface.

Syntax: Integer

Access: Read-only

Interface Group

locIfDelay

Provides the media-dependent delay in transferring a packet to another interface on the media. The delay is indicated in microseconds. Used by Interior Gateway Routing Protocol (IGRP).

Syntax: Integer

Access: Read-only

locIfDescr

Provides a description of the interface (such as Ethernet, serial, and so on) that corresponds to the user-configurable interface description commands.

Syntax: Display string

Access: Read-write

locIfFastInOctets

Provides the octet count for inbound traffic routed with fast and autonomous switching.

Syntax: Counter

Access: Read-only

locIfFastInPkts

Provides the packet count for inbound traffic routed with fast and autonomous switching.

Syntax: Counter

Access: Read-only

locIfFastOutOctets

Provides the octet count for outbound traffic routed with fast and autonomous switching.

Syntax: Counter

Access: Read-only

locIfFastOutPkts

Provides the packet count for outbound traffic routed with fast and autonomous switching.

Syntax: Counter

Access: Read-only

locIfHardType

Provides the type of interface (such as Ethernet, serial, FDDI, and so on).

Syntax: Display string

Access: Read-only

locIfInAbort

Provides the number of input packets that were aborted. Aborted input packets usually indicate a clocking problem between the serial interface and the data-link equipment.

Syntax: Integer

Access: Read-only

locIfInBitsSec

Provides a weighted 5-minute exponentially decaying average of interface input bits per second.

Syntax: Integer

Access: Read-only

locIfInCRC

Provides the number of input packets that had cyclic redundancy checksum (CRC) errors. The CRC generated by the originating station or far-end device does not match the checksum calculated from the data received. On a serial link, CRCs usually indicate noise, gain hits, or other transmission problems on the data link.

Syntax: Integer

Access: Read-only

locIfInFrame

Provides the number of input packets that were received incorrectly with framing errors. On a serial line, this is usually the result of noise or other transmission problems.

Syntax: Integer

Access: Read-only

locIfInGiants

Provides the number of input packets that were discarded because they exceeded the maximum packet size allowed by the physical media.

Syntax: Integer

Access: Read-only

locIfInIgnored

Provides the number of input packets that were ignored by this interface because the interface hardware ran low on internal buffers. Broadcast storms and bursts of noise can cause the ignored count to be increased.

Syntax: Integer

Access: Read-only

locIfInKeep

Indicates whether keepalives are enabled on this interface.

Syntax: Integer (1 = enabled, 2 = disabled)

Access: Read-only

locIfInOverrun

Provides the number of times the serial receiver hardware was unable to send data to a hardware buffer because the input rate exceeded the ability of the receiver to handle the data.

Syntax: Integer

Access: Read-only

locIfInPktsSec

Provides a weighted 5-minute exponentially decaying average of input packets.

Syntax: Integer

Access: Read-only

locIfInputQueueDrops

Provides the number of packets dropped because the input queue was full.

Syntax: Integer

Access: Read-only

Interface Group

locIfInRunts

Provides the number of input packets that were discarded because they were smaller than the minimum packet size allowed by the physical media.

Syntax: Integer

Access: Read-only

locIfLastIn

Provides the elapsed time in milliseconds since the last line protocol input packet was successfully received by an interface. Useful for knowing when a dead interface failed.

Syntax: Integer

Access: Read-only

locIfLastOut

Provides the elapsed time in milliseconds since the last line protocol output packet was successfully transmitted by an interface. Useful for knowing when a dead interface failed.

Syntax: Integer

Access: Read-only

locIfLastOutHang

Provides the elapsed time in milliseconds since the last line protocol output packet could not be successfully transmitted.

OR

Provides the elapsed time (in milliseconds) since the interface was last reset because of a transmission that took too long.

Syntax: Integer

Access: Read-only

locIfLineProt

Indicates whether the interface is up or down.

Syntax: Integer (1 = up, 2 = down)

Access: Read-only

locIfLoad

Provides the loading factor of the interface. The load on the interface is calculated as an exponential average over 5 minutes and expressed as a fraction of 255 (255/255 is completely saturated). Used by Interior Gateway Routing Protocol (IGRP).

Syntax: Integer

Access: Read-only

locIfOutBitsSec

Provides a weighted 5-minute exponentially decaying average of interface output bits per second for the specific protocol.

Syntax: Integer

Access: Read-only

locIfOutPktsSec

Provides a weighted 5-minute exponentially decaying average of interface output packets per second for the specific protocol.

Syntax: Integer

Access: Read-only

Interface Group

locIfOutputQueueDrops

Provides the number of packets dropped because the output queue was full.

Syntax: Integer

Access: Read-only

locIfReason

Provides the reason for the most recent status change of the interface.

Syntax: Display string

Access: Read-only

locIfReliab

Provides the level of reliability for the interface. The reliability of the interface is calculated as an exponential average over 5 minutes and expressed as a fraction of 255 (255/255 is 100 percent). Used by Interior Gateway Routing Protocol (IGRP).

Syntax: Integer

Access: Read-only

locIfResets

Provides the number of times the interface was reset internally. An interface can be reset if packets queued for transmission were not sent within several seconds. On a serial line, this can be caused by a malfunctioning modem that is not supplying the transmit clock signal or by a cable problem. If the system notices that the carrier detect line of a serial interface is up, but the line protocol is down, it periodically resets the interface in an effort to restart it. Interface resets also can occur when an interface is looped back or shut down.

Syntax: Integer

Access: Read-only

locIfRestarts

Provides the number of times the interface needed to be completely restarted because of errors.

Syntax: Integer

Access: Read-only

locIfSlowInOctets

Provides the octet count for inbound traffic routed with process switching.

Syntax: Counter

Access: Read-only

locIfSlowInPkts

Provides the packet count for inbound traffic routed with process switching.

Syntax: Counter

Access: Read-only

locIfSlowOutPkts

Provides the packet count for outbound traffic routed with process switching.

Syntax: Counter

Access: Read-only

locIfSlowOutOctets

Provides the octet count for outbound traffic routed with process switching.

Syntax: Counter

Access: Read-only

End of Table

Address Resolution Protocol (ARP)

The following variables in the Interface group apply to interfaces running the Address Resolution Protocol (ARP). ARP provides dynamic addressing between 32-bit IP addresses and Ethernet addresses. For detailed information on ARP, refer to the Cisco IOS configuration guides and command references.

locIfarpInOctets

Provides the ARP input octet count.

Syntax: Counter

Access: Read-only

locIfarpInPkts

Provides the ARP input packet count. It indicates the number of ARP Reply packets received by this router on this interface from other hosts.

Syntax: Counter

Access: Read-only

locIfarpOutOctets

Provides the ARP output octet count.

Syntax: Counter

Access: Read-only

locIfarpOutPkts

Provides the ARP output packet count. It indicates the number of ARP Request packets sent by this router on this interface to other hosts on the network.

Syntax: Counter

Access: Read-only

AppleTalk

The following variables in the Interface group apply to interfaces running AppleTalk.

locIfappletalkInOctets

Provides the AppleTalk protocol input octet count.

Syntax: Counter

Access: Read-only

locIfappletalkInPkts

Provides the AppleTalk protocol input packet count.

Syntax: Counter

Access: Read-only

locIfappletalkOutOctets

Provides the AppleTalk protocol output octet count.

Syntax: Counter

Access: Read-only

locIfappletalkOutPkts

Provides the AppleTalk protocol output packet count.

Syntax: Counter

Access: Read-only

Apollo

The following variables in the Interface group apply to interfaces running Apollo.

locIfapolloInOctets

Provides the Apollo protocol input octet count.

Syntax: Counter

Access: Read-only

locIfapolloInPkts

Provides the Apollo protocol input packet count.

Syntax: Counter

Access: Read-only

locIfapolloOutOctets

Provides the Apollo protocol output octet count.

Syntax: Counter

Access: Read-only

locIfapolloOutPkts

Provides the Apollo protocol output packet count.

Syntax: Counter

Access: Read-only

Bridging

The following variables in the Interface group apply to interfaces running bridging protocols.

locIfbridgedInOctets

Provides the bridged protocol input octet count.

Syntax: Counter

Access: Read-only

locIfbridgedInPkts

Provides the bridged protocol input packet count.

Syntax: Counter

Access: Read-only

locIfbridgedOutOctets

Provides the bridged protocol output octet count.

Syntax: Counter

Access: Read-only

locIfbridgedOutPkts

Provides the bridged protocol output packet count.

Syntax: Counter

Access: Read-only

Interface Group

locIfrsbInOctets

Provides the Source-Route Bridging (SRB) protocol input octet count.

Syntax: Counter

Access: Read-only

locIfrsbInPkts

Provides the SRB protocol input packet count.

Syntax: Counter

Access: Read-only

locIfrsbOutOctets

Provides the SRB protocol output octet count.

Syntax: Counter

Access: Read-only

locIfrsbOutPkts

Provides the SRB protocol output packet count.

Syntax: Counter

Access: Read-only

Connectionless Network Service (CLNS)

The following variables in the Interface group apply to interfaces running Connectionless Network Service (CLNS).

locIfclnsInOctets

Provides the CLNS protocol input byte count.

Syntax: Counter

Access: Read-only

locIfclnsInPkts

Provides the CLNS protocol input packet count.

Syntax: Counter

Access: Read-only

locIfclnsOutOctets

Provides the CLNS protocol output byte count.

Syntax: Counter

Access: Read-only

locIfclnsOutPkts

Provides the CLNS protocol output packet count.

Syntax: Counter

Access: Read-only

DECnet

The following variables in the Interface group apply to interfaces running DECnet.

locIfdecnetInOctets

Provides the DECnet protocol input octet count.

Syntax: Counter

Access: Read-only

locIfdecnetInPkts

Provides the DECnet protocol input packet count.

Syntax: Counter

Access: Read-only

locIfdecnetOutOctets

Provides the DECnet protocol output octet count.

Syntax: Counter

Access: Read-only

locIfdecnetOutPkts

Provides the DECnet protocol output packet count.

Syntax: Counter

Access: Read-only

HP Probe

The following variables in the Interface group apply to interfaces running HP Probe, an address resolution protocol developed by Hewlett-Packard.

locIfprobeInOctets

Provides the HP Probe protocol input octet count.

Syntax: Counter

Access: Read-only

locIfprobeInPkts

Provides the HP Probe protocol input packet count.

Syntax: Counter

Access: Read-only

locIfprobeOutOctets

Provides the HP Probe protocol output octet count.

Syntax: Counter

Access: Read-only

locIfprobeOutPkts

Provides the HP Probe protocol output packet count.

Syntax: Counter

Access: Read-only

Internet Protocol (IP)

The following variables in the Interface group apply to interfaces running the Internet Protocol (IP).

locIfipInOctets

Provides the IP input octet count.

Syntax: Counter

Access: Read-only

locIfipInPkts

Provides the IP input packet count.

Syntax: Counter

Access: Read-only

locIfipOutOctets

Provides the IP output octet count.

Syntax: Counter

Access: Read-only

locIfipOutPkts

Provides the IP output packet count.

Syntax: Counter

Access: Read-only

LAN Network Manager (LNM)

The following variables in the Interface group apply to interfaces running the LAN Network Manager (LNM) protocol. This protocol manages source-route bridging (SRB) networks.

locIfIflanmanInOctets

Provides the LAN Network Manager protocol input octet count.

Syntax: Counter

Access: Read-only

locIfIflanmanInPkts

Provides the LAN Network Manager protocol input packet count.

Syntax: Counter

Access: Read-only

locIfIflanmanOutOctets

Provides the LAN Network Manager protocol output octet count.

Syntax: Counter

Access: Read-only

locIfIflanmanOutPkts

Provides the LAN Network Manager protocol output packet count.

Syntax: Counter

Access: Read-only

Maintenance Operation Protocol (MOP)

The following variables in the Interface group apply to interfaces running the Maintenance Operation Protocol (MOP).

locIfmopInOctets

Provides the MOP input octet count.

Syntax: Counter

Access: Read-only

locIfmopInPkts

Provides the MOP input packet count.

Syntax: Counter

Access: Read-only

locIfmopOutOctets

Provides the MOP output octet count.

Syntax: Counter

Access: Read-only

locIfmopOutPkts

Provides the MOP output packet count.

Syntax: Counter

Access: Read-only

Novell

The following variables in the Interface group apply to interfaces running Novell.

locIfnovellInOctets

Provides the Novell protocol input octet count.

Syntax: Counter

Access: Read-only

locIfnovellInPkts

Provides the Novell protocol input packet count.

Syntax: Counter

Access: Read-only

locIfnovellOutOctets

Provides the Novell protocol output octet count.

Syntax: Counter

Access: Read-only

locIfnovellOutPkts

Provides the Novell protocol output packet count.

Syntax: Counter

Access: Read-only

Other Protocols

The following variables in the Interface group record the number of input and output packets and octets for interfaces running protocols other than those listed in the Interface group.

locIfotherInOctets

Provides the input octet count for protocols other than those listed in the Interface group.

Syntax: Counter

Access: Read-only

locIfotherInPkts

Provides the input packet count for protocols other than those listed in the Interface group.

Syntax: Counter

Access: Read-only

locIfotherOutOctets

Provides the output octet count for protocols other than those listed in the Interface group.

Syntax: Counter

Access: Read-only

locIfotherOutPkts

Provides the output packet count for protocols other than those listed in the Interface group.

Syntax: Counter

Access: Read-only

Serial Tunnel (STUN)

The following variables in the Interface group apply to interfaces using the Serial Tunnel (STUN) protocol. STUN allows devices that use Synchronous Data Link Control (SDLC) or High-Level Data Link Control (HDLC) to be connected through one or more Cisco routers across different network topologies.

locIfstunInOctets

Provides the STUN protocol input octet count.

Syntax: Counter

Access: Read-only

locIfstunInPkts

Provides the STUN protocol input packet count.

Syntax: Counter

Access: Read-only

locIfstunOutOctets

Provides the STUN protocol output octet count.

Syntax: Counter

Access: Read-only

locIfstunOutPkts

Provides the STUN protocol output packet count.

Syntax: Counter

Access: Read-only

Spanning Tree

The following variables in the Interface group apply to interfaces running the Spanning Tree protocol. Used in bridging, spanning trees provide root and designated bridges to notify all other bridges in the network when an address change has occurred, thereby eliminating loops.

locIfspanInOctets

Provides the spanning-tree input octet packet count.

Syntax: Counter

Access: Read-only

locIfspanInPkts

Provides the spanning-tree input protocol packet count.

Syntax: Counter

Access: Read-only

locIfspanOutOctets

Provides the spanning-tree output octet packet count.

Syntax: Counter

Access: Read-only

locIfspanOutPkts

Provides the spanning-tree output protocol packet count.

Syntax: Counter

Access: Read-only

Banyan Virtual Integrated Network Service (VINES)

The following variables in the Interface group apply to interfaces running the Banyan Virtual Integrated Network Service (VINES) protocol. This proprietary protocol is derived from the Xerox Network Systems (XNS) protocol. The VINES variables provide the number of input and output packets and octets on a per interface basis.

locIfvinesInOctets

Provides the VINES protocol input octet count.

Syntax: Counter

Access: Read-only

locIfvinesInPkts

Provides the VINES protocol input packet count.

Syntax: Counter

Access: Read-only

locIfvinesOutOctets

Provides the VINES protocol output octet count.

Syntax: Counter

Access: Read-only

locIfvinesOutPkts

Provides the VINES protocol output packet count.

Syntax: Counter

Access: Read-only

Xerox Network Systems (XNS)

The following variables in the Interface group apply to interfaces running Xerox Network Systems (XNS).

locIfxnsInOctets

Provides the XNS protocol input octet count.

Syntax: Counter

Access: Read-only

locIfxnsInPkts

Provides the XNS input packet count.

Syntax: Counter

Access: Read-only

locIfxnsOutOctets

Provides the XNS protocol output octet count.

Syntax: Counter

Access: Read-only

locIfxnsOutPkts

Provides the XNS protocol output packet count.

Syntax: Counter

Access: Read-only

Internet Protocol (IP) Group

The Internet Protocol (IP) group provides variables pertaining to the IP, such as the determination of how an interface obtained its IP address, who supplied the address, and Internet Control Message Protocol (ICMP) messages about IP packet processing.

IP Address Table

The Cisco IP Address table, *lipAddrTable*, contains the following six variable entries, or rows: *locIPHelper*, *locIPHow*, *locIPRedirects*, *locIPSecurity*, *locIPUnreach*, and *locIPWho*. The index to this table is the IP address of the device, or *ipAdEntAddr*. If a device has *n* number of IP addresses, there will be *n* rows in the table.

For simplification, Table 2-4 shows only the *locIpHow* and *locIPWho* variables. The *locIPHow* variable value shows that the device at 131.108.201.245 obtained its address through nonvolatile memory. The *locIPWho* variable value indicates the device was assigned its current address by the device at 131.101.200.248.

Table 2-4 IP Address

IP Address	locIPHow	locIPWho	and so on
131.108.201.245	nonvolatile	131.101.200.248	
142.111.202.244	nonvolatile	131.56.70.249	
and so on			

locIPHelper

Provides the IP address for broadcast forwarding support. Provides the destination broadcast or IP address that the router should use when forwarding User Datagram Protocol (UDP) broadcast datagrams, including BootP, received on the interface.

Syntax: IpAddress

Access: Read-only

locIPHow

Describes how this interface obtained its IP address. Typically, the address is determined by nonvolatile memory.

Syntax: Display string

Access: Read-only

locIPRedirects

Indicates whether Internet Control Message Protocol (ICMP) redirects will be sent. A router sends an ICMP Redirect message to the originator of any datagram that it is forced to resend through the same interface on which it was received. It does so because the originating host presumably could have sent that datagram to the ultimate destination without involving the router at all. ICMP Redirect messages are sent only if the router is configured with the **ip redirects** command.

Syntax: Integer (1 = sent, 2 = not sent)

Access: Read-only

locIPSecurity

Indicates whether IP security is enabled on the interface. For details on IP security levels, see RFC 1108, *U.S. Department of Defense Security Options for the Internet Protocol*.

Syntax: Integer (0 = false, 1 = true)

Access: Read-only

locIPUnreach

Indicates whether Internet Control Message Protocol (ICMP) packets indicating unreachable addresses will be sent for a specific route.

If this variable is set, and the router receives a datagram that it cannot deliver to its ultimate datagram (because it knows of no route to the destination address), it replies to the originator of that datagram with an ICMP Host Unreachable message.

Syntax: Integer (0 = false, 1 = true)

Access: Read-only

locIPWho

Provides the IP address of the device from which this interface received its IP address. If the interface does not use an IP address from another device, a value of 0.0.0.0 displays.

Syntax: IPAddress

Access: Read-only

End of Table

IP Routing Table

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

In Table 2-5, for the route with the destination IP address of 131.104.111.1, the routing table network mask is 255.255.255.0. The number of parallel routes within the routing table is 3, and the route was used in a forwarding operation two times.

Table 2-5 **IP Routing Table**

ipRouteDest	locRtMask	locRtCount
131.104.111.1	255.255.255.0	3
133.45.244.245	255.255.255.0	1

Internet Protocol (IP) Group

locRtCount

Provides the number of parallel routes within the IP Routing table.

Syntax: Integer

Access: Read-only

locRtMask

Provides the IP Routing table network mask. For example, 255.255.255.0.

Syntax: IpAddress

Access: Read-only

End of Table

actLostByts

Provides the total number of bytes of lost IP packets as a result of accounting failure.

Syntax: Integer

Access: Read-only

actLostPkts

Provides the number of IP packets that were lost due to memory limitations and accounting failure.

Syntax: Integer

Access: Read-only

actThresh

Provides the threshold of IP accounting records in use before IP traffic will be discarded.

Syntax: Integer

Access: Read-only

IP Accounting Group

Cisco routers maintain two accounting databases: an active database and a checkpoint database. The router takes a snapshot of the running, or active database, and copies it into the checkpoint database. For detailed information on active and checkpoint databases, refer to the Cisco IOS configuration guides and command references.

This group provides access to the active database that is created and maintained if IP accounting is enabled on a router. The active database contains information about the number of bytes and packets switched through a system on a source and destination IP address basis. Only transit IP traffic is measured and only on an outbound basis; traffic generated by the router or terminating in the router is not included in the accounting statistics. Internetwork statistics obtained through these variables can be analyzed to improve network performance.

IP Accounting Table

The local IP accounting table, *lipAccountingTable*, includes four related variables: *actByts*, *actDst*, *actPkts*, and *actSrc*. The index for this table is *actSrc* and *actDst*. For example, in the first row in Table 2-6, the source host address is 131.24.35.248, and the destination host address is 138.32.28.245. Fifty IP packets and 400 bytes of data have been sent between the source and destination address.

Table 2-6 Local IP Accounting Table

actByts	actDst	actPkts	actSrc
400	138.32.28.245	50	131.24.35.248
1259	128.52.33.101	110	128.52.33.96

actByts

Provides the total number of bytes in IP packets from the source to destination host.

Syntax: Integer

Access: Read-only

actDst

Provides the IP destination address for the host traffic matrix.

Syntax: Ip Address

Access: Read-only

actPkts

Provides the number of IP packets sent from the source to destination host.

Syntax: Integer

Access: Read-only

actSrc

Provides the IP address for the host traffic matrix.

Syntax: IPAddress

Access: Read-only

actViolation

Specifies the access list number violated by packets from this source to this destination. A zero value indicates that no access list was violated.

Syntax: Integer

Access: Read-only

End of Table

actAge

Provides the age of the accounting data in the current data matrix of the active database.

Syntax: Timeticks

Access: Read-only

IP Checkpoint Accounting Group

The Cisco router maintains two accounting databases: an active database and a checkpoint database. For detailed information on active and checkpoint databases, refer to the Cisco IOS configuration guides and command references.

The running, or active database, is copied into the checkpoint database. If the checkpoint database already has data obtained previously from the active database, the router appends the latest copy of the active database to the existing data in the checkpoint database. The checkpoint database stores data retrieved from the active database until `actCheckPoint` is set or you delete the contents of this database by using the **clear ip accounting [checkpoint]** command.

A network management system (NMS) can use checkpoint MIB variables to analyze stable data in the checkpoint database.

IP Checkpoint Accounting Table

The local IP Checkpoint Accounting table, *lipCkAccountingTable*, includes four related variables: *ckactByts*, *ckactDst*, *ckactPkts*, and *ckactSrc*. The index for this table is *ckacSrc* and *ckactDst*. For example, in Table 2-7, the source host address is 131.24.35.248. The destination host address is 138.32.28.245. Fifty IP packets and 400 bytes of data have been sent between the source and destination address.

Table 2-7 IP Checkpoint Accounting

ckactByts	ckactDst	ckactPkts	ckacSrc
400	138.32.28.245	50	131.24.35.248
480	124.45.222.246	60	123.34.216.244

ckactByts

Provides the total number of bytes in IP packets from source to destination in the checkpoint matrix.

Syntax: Integer

Access: Read-only

ckactDst

Provides the IP destination address of the host receiving the IP packets. The address is listed in the checkpoint traffic matrix.

Syntax: IpAddress

Access: Read-only

ckactPkts

Provides the number of IP packets sent from the source to the destination address in the checkpoint matrix.

Syntax: Integer

Access: Read-only

ckactSrc

Provides the IP source address of the host sending the IP packets. The address is listed in the checkpoint traffic matrix.

Syntax: IP address

Access: Read-only

ckactViolation

Provides the access list number violated by packets from source to destination in the checkpoint matrix.

Syntax: Integer

Access: Read-only

End of Table

actCheckPoint

Activates a checkpoint database. This variable must be read and then set to the same value that was read. The value read and then set will be incremented after a successful set request.

For detailed information on active and checkpoint databases, refer to the Cisco IOS configuration guides and command references.

Syntax: Integer

Access: Read-write

ckactAge

Provides information on how long ago the data was first stored in the checkpoint matrix.

Syntax: Timeticks

Access: Read-only

ipNoaccess

Provides the total number of packets dropped due to access control failure.

Syntax: Counter

Access: Read-only

System Group

The variables described in this section are system-wide and apply to all Cisco Systems products.

Basic

The following variables pertain to basic information such as system software description and version number, host and domain names, and number of bytes of free memory in the managed device.

authAddr

Provides the IP address of the device causing the last SNMP authorization failure. The device did not use a configured community string or tried a SET with a read-only community string.

Syntax: IP address

Access: Read-only

bootHost

Provides the IP address of the host that supplied the software currently running on the managed device.

Syntax: IP address

Access: Read-only

domainName

Provides the domain portion of the domain name of the host.

Syntax: Display string

Access: Read-only

hostName

This variable represents the name of the host in printable ASCII characters.

Syntax: DisplayString

Access: Read-only

hostName

Represents the name of the host in printable ASCII characters.

Syntax: Display string

Access: Read-only

romId

Contains a printable octet string that contains the system bootstrap description and version identification.

Syntax: Display string

Access: Read-only

whyReload

Contains a printable octet string that contains the reason why the system was last restarted.

Syntax: Display string

Access: Read-only

Buffer

The following variables are used to monitor the amount and type of buffer space available within a managed device. *Buffers* are blocks of memory used to hold network packets. There are five types of buffers based on size: *small*, *middle*, *big*, *large*, and *huge*. There are

System Group

several pools of different-sized buffers. These pools grow and shrink based upon demand. Some buffers are temporary and are created and destroyed as warranted. Others are permanently allocated.

bufferFail

Contains the total number of allocation requests that have failed due to lack of any free buffers.

Syntax: Integer

Access: Read-only

bufferNoMem

Counts the number of failures caused by insufficient memory to create a new buffer.

Syntax: Integer

Access: Read-only

Buffer Elements

Buffer elements are blocks of memory used in internal operating system queues.

bufferElCreate

Contains the number of new buffer elements created for the managed device.

Syntax: Integer

Access: Read-only

bufferElFree

Contains the number of buffer elements that are not currently allocated and are available for use in the managed device.

Syntax: Integer

Access: Read-only

bufferElHit

Contains the number of successful attempts to allocate a buffer element when needed.

Syntax: Integer

Access: Read-only

bufferElMax

Contains the maximum number of buffer elements the managed device can have.

Syntax: Integer

Access: Read-only

bufferElMiss

Contains the number of allocation attempts that failed because there were no buffer elements available.

Syntax: Integer

Access: Read-only

freeMem

Provides the number of bytes of free memory available in the managed device.

Syntax: Integer

Access: Read-only

Small Buffers

Small buffer sizes are configurable.

bufferSmCreate

Contains the number of small buffers created in the managed device.

Syntax: Integer

Access: Read-only

bufferSmFree

Contains the number of small buffers that are currently available to the managed device.

Syntax: Integer

Access: Read-only

bufferSmHit

Contains the number of successful attempts to allocate a small buffer when needed.

Syntax: Integer

Access: Read-only

bufferSmMax

Contains the maximum number of small buffers that can be allocated to the managed device.

Syntax: Integer

Access: Read-only

bufferSmMiss

Contains the number of allocation attempts that failed because there were no small buffers available.

Syntax: Integer

Access: Read-only

bufferSmSize

Provides the size (in bytes) of small buffers.

Syntax: Integer

Access: Read-only

bufferSmTotal

Provides the total number of small buffers allocated to the managed device.

Syntax: Integer

Access: Read-only

bufferSmTrim

Contains the small buffers that have been destroyed in the managed device.

Syntax: Integer

Access: Read-only

Middle Buffers

Middle buffer sizes are configurable.

bufferMdCreate

Contains the number of middle buffers created in the managed device.

Syntax: Integer

Access: Read-only

bufferMdFree

Contains the number of middle buffers that are currently available to the managed device.

Syntax: Integer

Access: Read-only

bufferMdHit

Contains the number of successful attempts to allocate a middle buffer when needed.

Syntax: Integer

Access: Read-only

bufferMdMax

Contains the maximum number of middle buffers that can be allocated to the managed device.

Syntax: Integer

Access: Read-only

bufferMdMiss

Contains the number of allocation attempts that failed because there were no middle buffers available.

Syntax: Integer

Access: Read-only

bufferMdSize

Provides the size (in bytes) of middle buffers.

Syntax: Integer

Access: Read-only

bufferMdTotal

Provides the total number of middle buffers allocated to the managed device.

Syntax: Integer

Access: Read-only

bufferMdTrim

Contains the middle buffers that have been destroyed in the managed device.

Syntax: Integer

Access: Read-only

Big Buffers

Big buffer sizes are configurable.

bufferBgCreate

Contains the number of big buffers created in the managed device.

Syntax: Integer

Access: Read-only

bufferBgFree

Contains the number of big buffers that are currently available to the managed device.

Syntax: Integer

Access: Read-only

bufferBgHit

Contains the number of successful attempts to allocate a big buffer when needed.

Syntax: Integer

Access: Read-only

bufferBgMax

Contains the maximum number of big buffers that can be allocated to the managed device.

Syntax: Integer

Access: Read-only

bufferBgMiss

Contains the number of allocation attempts that failed because there were no big buffers available.

Syntax: Integer

Access: Read-only

bufferBgSize

Provides the size (in bytes) of big buffers.

Syntax: Integer

Access: Read-only

bufferBgTotal

Provides the total number of big buffers allocated to the managed device.

Syntax: Integer

Access: Read-only

bufferBgTrim

Contains the big buffers that have been destroyed in the managed device.

Syntax: Integer

Access: Read-only

Large Buffers

Large buffer sizes are configurable.

bufferLgCreate

Contains the number of large buffers created in the managed device.

Syntax: Integer

Access: Read-only

bufferLgFree

Contains the number of large buffers that are currently available to the managed device.

Syntax: Integer

Access: Read-only

bufferLgHit

Contains the number of successful attempts to allocate a large buffer when needed.

Syntax: Integer

Access: Read-only

bufferLgMax

Contains the maximum number of large buffers that can be allocated to the managed device.

Syntax: Integer

Access: Read-only

bufferLgMiss

Contains the number of allocation attempts that failed because there were no large buffers available.

Syntax: Integer

Access: Read-only

bufferLgSize

Provides the size (in bytes) of large buffers.

Syntax: Integer

Access: Read-only

bufferLgTotal

Provides the total number of large buffers allocated to the managed device.

Syntax: Integer

Access: Read-only

bufferLgTrim

Contains the large buffers that have been destroyed in the managed device.

Syntax: Integer

Access: Read-only

Huge Buffers

Huge buffer sizes are configurable.

bufferHgCreate

Contains the number of huge buffers created in the managed device.

Syntax: Integer

Access: Read-only

bufferHgFree

Contains the number of huge buffers that are currently available to the managed device.

Syntax: Integer

Access: Read-only

bufferHgHit

Contains the number of successful attempts to allocate a huge buffer when needed.

Syntax: Integer

Access: Read-only

bufferHgMax

Contains the maximum number of huge buffers that can be allocated to the managed device.

Syntax: Integer

Access: Read-only

bufferHgMiss

Contains the number of allocation attempts that failed because there were no huge buffers available.

Syntax: Integer

Access: Read-only

bufferHgSize

Provides the size (in bytes) of huge buffers.

Syntax: Integer

Access: Read-only

bufferHgTotal

Provides the total number of huge buffers allocated to the managed device.

Syntax: Integer

Access: Read-only

bufferHgTrim

Contains the huge buffers that have been destroyed in the managed device.

Syntax: Integer

Access: Read-only

CPU Utilization

The following variables provide statistics on the CPU utilization of a device.

avgBusy1

Provides a cumulative average of the CPU usage percentage over a 1-minute period. This variable, called by the scheduler every 5 seconds, computes the busy time in the last 5-second period, and the 5-minute, exponentially decayed busy time. The following equation shows the average sampling time:

$$\text{average} = ((\text{average-interval}) * \exp(-t/C)) + \text{interval}$$

where t is 5 seconds and C is 1 minute, $\exp(-5/60) \approx .920 \approx 942/1024$

Syntax: Integer

Access: Read-only

avgBusy5

Provides a cumulative average of the CPU usage percentage over a 5-minute period. This variable, called by the scheduler every 5 seconds, computes the busy time in the last 5-second period, and the 5-minute, exponentially decayed busy time. The following equation shows the average sampling time:

$$\text{average} = ((\text{average-interval}) * \exp(-t/C)) + \text{interval}$$

where t is 5 seconds and C is five minutes, $\exp(-5/60*5) \approx .983 \approx 1007/1024$

Syntax: Integer

Access: Read-only

avgBusyPer

Provides the percentage of CPU usage over the first 5-second period in the scheduler. The scheduler determines which process or task takes priority over another and triggers them accordingly.

Syntax: Integer

Access: Read-only

ciscoContactInfo

Provides the Cisco name and address for reference purposes. This MIB variable applies only to router products that were purchased from Cisco.

Syntax: Display string

Access: Read-only

Environmental Monitor Card

The variables in this section have been deprecated and replaced with the variables in the Environmental Monitor group section above, also found in the ciscoMgmt tree.

The environmental card is provided only with the Cisco AGS+ router, however, Cisco IOS Release 11.1 does not support the Cisco AGS+ router. The Cisco 7000 router has built-in environmental monitoring functionality, and so does not use the card. The Cisco 7000 router provides environmental monitoring, reporting, and if necessary, system shutdown. However, a subset of the Environmental Monitor card variables apply to the Cisco 7000, and are listed below.

envPresent

Indicates whether there is an environmental monitor card in a router.

Syntax: Integer (0 = no, 1 = yes, but unavailable to SNMP; 2 = yes and available to SNMP for AGS+ routers; 3 = yes and available to SNMP for Cisco 7000 routers)

Access: Read-only

System Group

envTestPt1Descr

Test point 1 is the temperature of air entering the router.

Syntax: Display string

Access: Read-only

envTestPt1last

Provides the temperature of air entering the the Cisco 7000 router when the last shutdown occurred.

Syntax: Integer

Access: Read-only

envTestPt1MarginVal

Provides warning and fatal threshold values of the internal intake air for the Cisco 7000.

Syntax: Integer

Access: Read-only

envTestPt1Measure

Provides the current temperature of air entering the router.

Syntax: Display string

Access: Read-only

envTestPt1warn

Indicates whether the air temperature entering the router is at warning level.

Syntax: Integer (1 = warning, 2 = no warning)

Access: Read-only

envTestPt2Descr

Provides the temperature of air leaving the router.

Syntax: Display string

Access: Read-only

envTestPt2last

Provides the temperature of air leaving the router when the last shutdown occurred.

Syntax: Integer

Access: Read-only

envTestPt2MarginVal

Provides the fatal threshold value for the exhaust air flow of the router.

Syntax: Integer

Access: Read-only

envTestPt2Measure

Provides the temperature of the exhaust air flow of the router.

Syntax: Integer

Access: Read-only

envTestPt2warn

Indicates whether the temperature of air flow leaving the router is at a warning level.

Syntax: Integer (1 = warning, 2 = no warning)

Access: Read-only

System Group

envTestPt3Descr

Test point 3 is the +5-volt (V) line on the router.

Syntax: Display string

Access: Read-only

envTestPt3last

Provides the value of the +5V line when the last shutdown occurred.

Syntax: Integer

Access: Read-only

envTestPt3Measure

Provides the current value for the +5V line to the power supply on the router. The value is expressed in millivolts. (AGS+ and Cisco 7000)

Syntax: Integer

Access: Read-only

envTestPt3warn

Indicates whether the +5V line to the power supply is at warning level. The warning threshold is ± 5 percent above or below +5V.

Syntax: Integer (1 = warning, 2 = no warning)

Access: Read-only

envTestPt4Descr

Test point 4 is the +12V line to the power supply of the router.

Syntax: Display string

Access: Read-only

envTestPt4last

Provides the value of the +12V line when the last shutdown occurred.

Syntax: Integer

Access: Read-only

envTestPt4Measure

Provides the current value (in millivolts) of the +12V line to the power supply of the router.

Syntax: Integer

Access: Read-only

envTestPt4warn

Indicates whether the +12V line to the power supply is at warning level. The warning threshold is ± 10 percent above or below +12V.

Syntax: Integer (1 = warning, 2 = no warning)

Access: Read-only

envTestPt5Descr

Test point 5 is the -12V line to the power supply of the router.

Syntax: Display string

Access: Read-only

envTestPt5last

Provides the value of the -12V line when the last shutdown occurred.

Syntax: Integer

Access: Read-only

System Group

envTestPt5Measure

Provides the current value (in millivolts) of the -12V line to the power supply of the router.

Syntax: Integer

Access: Read-only

envTestPt5warn

Indicates whether the -12V line to the power supply on the router is at the warning level. The warning threshold is ± 10 percent above or below -12V. (AGS+ only)

Syntax: Integer (1 = warning, 2 = no warning)

Access: Read-only

envTestPt6Descr

Test point 6 is the +24V line to the power supply of the Cisco 7000 router.

Syntax: Display string

Access: Read-only

envTestPt6last

Provides the value of the +24V line to the power supply of the Cisco 7000 router when the last shutdown occurred.

Syntax: Integer

Access: Read-only

envTestPt6Measure

Provides the current value (in millivolts) of the +24V line to the power supply of the Cisco 7000 router.

Syntax: Integer

Access: Read-only

envTestPt6warn

Indicates whether the +24V line to the power supply of the Cisco 7000 router is at the warning level. The warning threshold is ± 10 percent above or below +24V.

Syntax: Integer (1 = warning, 2 = no warning)

Access: Read-only

Host Configuration File

The *hostConfig* variables are obsolete. Used to monitor and set host configuration file information, they consisted of the following:

- hostConfigAddr
- hostConfigName
- hostConfigProto
- hostConfigSet

Network Configuration File

The following variables are used to monitor and remotely set network configuration file information for the device.

System Group

netConfigAddr

Provides the address of the host that supplied the network configuration file for the managed device. The *network configuration file* contains commands that apply to all network servers and terminal services on a network.

Syntax: IPAddress

Access: Read-only

netConfigName

Provides the name of the network configuration file that resides on the managed device.

Syntax: Display string

Access: Read-only

netConfigProto

Provides the protocol that supplied the network configuration file.

Syntax: Integer

Access: Read-only

netConfigSet

Loads a new network configuration file via Trivial File Transfer Protocol (TFTP) onto the managed device and indicates the name of this configuration file. The instance ID is the IP address of the TFTP host. The display string indicates the name of the configuration file.

Syntax: Display string

Access: Write-only

System Configuration

The following variables are used to monitor and set system-wide parameters.

sysClearARP

Performs a clearing of the entire Address Resolution Protocol (ARP) cache and Internet Protocol (IP) route cache. The ARP provides dynamic mapping between IP addresses and Ethernet addresses. The ARP Cache table, which keeps a record of these mappings, can be cleared for maintenance purposes.

The IP route cache controls the use of a high-speed switching cache for IP routing. The route cache is enabled by default and allows outgoing packets to be load balanced on a per-destination basis. The *sysClearARP* variable helps clear the IP route cache for maintenance purposes.

Syntax: Integer

Access: Write-only

sysClearInt

Clears an interface that is given *IfIndex* as a value. To clear an interface, take the *ifIndex* for the interface (for example, a value of 4) and set the *sysClearInt* variable to the *ifIndex* value of 4.

Syntax: Integer

Access: Write-only

sysConfigAddr

Provides the address of the host that supplied the system boot image for the managed device. New versions of software can be downloaded over the network with boot image files. The new file takes effect the next time the managed device is reloaded.

Syntax: IPAddress

Access: Read-only

System Group

sysConfigName

Provides the name of the system boot image file. New versions of software can be downloaded over the network with boot image files. The new file takes effect the next time the managed device is reloaded.

Syntax: Display string

Access: Read-only

sysConfigProto

Provides the protocol type that supplied the system boot image.

Syntax: Integer

Access: Read-only

writeMem

Writes the current (running) router configuration into nonvolatile memory where it can be stored and retained even if the router is reloaded. Write configuration memory if 1. Erase configuration memory if 0.

Syntax: Integer

Access: Write-only

writeNet

Sends a copy of the current configuration via Trivial File Transfer Protocol (TFTP) to a remote host. When it is stored on the host, the configuration file can be edited and retrieved by other network entities.

Syntax: Display string

Access: Write-only

Terminal Services Group

Following are variables that can be applied to terminal services. This group of variables contains terminal service information on a per-line basis, such as line status, line type, line speed, type of flow control, and type of modem.

tsLines

Provides the number of physical lines on the device.

Syntax: Integer

Access: Read-only

tsClrTtyLine

Specifies the TTY line to clear. The read operation returns the last line cleared. A value of -1 indicates that no lines have been cleared.

Syntax: Integer

Access: Read-write

Terminal Services Line Table

The local terminal services line table, *ltsLineTable*, contains all the variables described in this section. The index to this table is the number of the terminal services line. If there are n number of terminal lines associated with the device, there will be n rows in the table.

Table 2-8 Terminal Services Line

Line Number	tsLineActive	tsLineAutobaud	and so on
1	Contains all of the variables described in this section.		
2			
and so on			

tsLineActive

Indicates whether this line is active.

Syntax: Integer (1 = active, 2 = not active)

Access: Read-only

tsLineAutobaud

Indicates whether the line is set to automatic baud rate detection so that it can adapt to the rate at which data is being sent to it.

Syntax: Integer (1 = autobaud, 2 = not autobaud)

Access: Read-only

tsLineEsc

Indicates what is used to represent the escape (Esc) character. The escape character allows a user to break out of active sessions.

Syntax: Display string

Access: Read-only

tsLineFlow

Indicates the type of flow control the line is using. The flow can be controlled from software or hardware. Input indicates that the flow control is coming from the device to the terminal service. Output indicates that flow control is provided by the terminal service.

The possible integer values follow:

1 = unknown

2 = none

3 = software-input

4 = software-output

5 = software-both

6 = hardware-input
7 = hardware-output
8 = hardware-both

Syntax: Integer

Access: Read-only

tsLineLoc

Describes the physical location of the line. The integer values 1 through 3 represent commands that can be defined by the user.

Syntax: Display string

Access: Read-only

tsLineModem

Describes the type of modem control the line is using.

The possible integer values follow:

1 = unknown
2 = none
3 = call-in
4 = call-out
5 = cts-required
6 = ri-is-cd
7 = modem inout

Descriptions of the integer values follow:

Call-in indicates dial-in modems that use the status of Data Terminal Ready (DTR) to determine whether to answer an incoming call.

Call-out indicates modems that raise data terminal ready (DTR) to see if Clear To Send (CTS) becomes high as an indication that the host has noticed its signal.

Cts-required indicates the form of modem control that requires CTS to be high throughout the use of the line.

ri-is-cd is used for lines with high-speed modems. The modem answers the call if DTR is high, uses its Carrier Detect (CD) signal to reflect the carrier presence, and has its CD signal wired to the ring input of the terminal service.

modem inout is used to configure a line for both incoming and outgoing calls. The command enables a line to be used for both incoming and outgoing calls on dial-in/dial-out modems.

Syntax: Integer

Access: Read-only

tsLineNoise

Provides the number of garbage characters received while the line is inactive.

Syntax: Integer

Access: Read-only

tsLineNses

Indicates the number of current sessions on the line.

Syntax: Integer

Access: Read-only

tsLineNumber

The terminal session line number.

Syntax: Integer

Access: Read-only

tsLineRotary

Specifies the number of the rotary group to which the line belongs. If the first line in a rotary group is busy, a connection can be made to the next free line.

Syntax: Integer

Access: Read-only

tsLineScrLen

Provides the length (in lines) of the screen of the terminal attached to the line.

Syntax: Integer

Access: Read-only

tsLineScrwid

Provides the width (in characters) of the screen of the terminal attached to the line.

Syntax: Integer

Access: Read-only

tsLineSestmo

Specifies the interval (in seconds) for closing the connection when there is no input or output traffic during a session.

Syntax: Integer

Access: Read-only

tsLineSpeedin

Indicates the input speed at which the line is running.

Syntax: Integer

Access: Read-only

tsLineSpeedout

Indicates the output speed at which the line is running.

Syntax: Integer

Access: Read-only

tsLineTerm

Describes the terminal type of the line.

Syntax: Display string

Access: Read-only

tsLineTimeActive

The time in seconds since the line was activated.

Syntax: Integer

Access: Read-only

tsLineTmo

Specifies the interval (in seconds) for closing the connection when there is no input or output traffic on the line.

Syntax: Integer

Access: Read-only

tsLineType

Describes the terminal line type.

The possible integer values follow:

- 1 = unknown
- 2 = console
- 3 = terminal
- 4 = line-printer
- 5 = virtual-terminal
- 6 = auxiliary

Syntax: Integer

Access: Read-only

tsLineUser

Provides the Terminal Access Controller Access Control System (TACACS) username and indicates whether TACACS is enabled on this line. TACACS servers provide security for accessing terminals remotely.

Syntax: Display string

Access: Read-only

tsLineUses

Indicates the number of times a connection has been made to or from this line.

Syntax: Integer

Access: Read-only

End of Table

Terminal Services Line Session Table

The Terminal Services Line Session table, *tsLineSessionTable*, contains eight variables: *tslineSesAddr*, *tslineSesCur*, *tslineSesDir*, *tslineSesIdle*, *tslineSesLine*, *slineSesName*, *tslineSesSession*, and *slineSesType*.

For simplification, Table 2-9 shows values for three of the variables contained in the Terminal Services Line Session table. The index to the table is the session number and line number. Line 1 in the first session illustrates a Telnet connection. The session was started by the terminal. The remote host for this session is located at the IP address of 131.38.141.244.

Table 2-9 Terminal Services Line Session

Session no. Line no.	tslineSesAddr	tslineSesDir	tslineSesType
1, 1	131.38.141.244	3	5
2, 4	138.121.128.243	2	3

tslineSesAddr

Provides the address of the remote host for this session.

Syntax: Network address

Access: Read-only

tslineSesCur

Indicates whether this session is currently active.

Syntax: Integer (1 = active, 2 = not active)

Access: Read-only

tslineSesDir

Indicates whether this session was started by another device (incoming) or by the terminal (outgoing).

The possible integer values follow:

1 = unknown

2 = incoming

3 = outgoing

Syntax: Integer

Access: Read-only

tslineSesIdle

Indicates the amount of time (in seconds) that this session has been idle.

Syntax: Integer

Access: Read-only

tslineSesLine

Table index 1.

Syntax: Integer

Access: Read-only

tslineSesName

Provides the name of the remote host for this session.

Syntax: Display string

Access: Read-only

tslineSesSession

Table index 2.

Syntax: Integer

Access: Read-only

tslineSesType

Describes the type of session that is currently active.

The possible integer values follow:

- 1 = unknown
- 2 = X.3 Packet Assembler/Disassembler (PAD)
- 3 = stream (enables a raw TCP stream with no Telnet-control sequences)
- 4 = rlogin (for making remote connection to a host—part of TCP/IP)
- 5 = telnet (for making remote connection to a host—UNIX protocol)
- 6 = Transmission Control Protocol (TCP)
- 7 = local-area transport (LAT)
- 8 = Maintenance Operation Protocol (MOP)
- 9 = Serial Line Internet Protocol (SLIP)
- 10 = XRemote (provides support for X Windows over a serial line)

Syntax: Integer

Access: Read-only

End of Table

Terminal Services Messages

The following variables pertain to the parameters of terminal services messages.

tsMsgDuration

Sets the length of time (in milliseconds) allocated to reissue a message. The minimum nonzero setting is 10000.0. A setting of 0 will not repeat the message.

Syntax: Integer

Access: Read-write

tsMsgIntervaltim

Sets the interval (in milliseconds) that occurs between reissues of the same message. The minimum (nonzero) setting for this interval is 10,000 milliseconds. If set to 0, the intervals will become more frequent as the message duration gets close to expiring. For example, 2 hours, 1 hour, 30 minutes, 5 minutes, and 1 minute.

Syntax: Integer

Access: Read-write

tsMsgSend

Determines what action to take after the message has been sent.

The possible integer values follow:

1 = nothing

2 = reload

3 = message done

4 = abort

Syntax: Integer

Access: Read-write

Transmission Control Protocol (TCP) Group

tsMsgText

Sets the text of the message. Up to 256 characters can be included in the message.

Syntax: Display string

Access: Read-write

tsMsgTmpBanner

Determines whether to use the message text as a temporary banner.

Syntax: Integer (1 = no, 2 = yes, in addition to the regular banner)

Access: Read-write

tsMsgTtyLine

Selects the TTY line to which you want the message sent. Setting this variable to -1 will send the message to all TTY lines.

Syntax: Integer

Access: Read-write

Transmission Control Protocol (TCP) Group

The following variables, from the TCP group in Cisco IOS Release 10.2, have been deprecated and replaced with the Cisco Transmission Control Protocol (ciscoTCP) group, found in the ciscoMgmt tree.

These variables can be applied to Cisco products running the Transmission Control Protocol (TCP). These variables provide statistics on the number of input and output bytes and packets for TCP connections.

TCP Connection Table

The TCP connection table, *ltpConnTable*, contains five variables: *loctcpConnElapsed*, *loctcpConnInBytes*, *loctcpConnInPkts*, *loctcpConnOutBytes*, and *loctcpConnOutPkts*.

The index to this table includes the local host address and port number and the remote host address and port number for each TCP connection that is active for the device. These values are represented by *tcpConnLocalAddress*, *tcpConnLocalPort*, *tcpConnRemAddress*, and *tcpConnRemPort* from RFC 1213.

For n number of TCP connections, there are n rows in the table. The value n can change at any time if another TCP connection opens or if an existing TCP connection closes.

In Table 2-10, TCP A represents the first TCP connection in the table. The TCP A connection shows 100 input bytes, 100 output bytes, 85 input packets, and 85 output packets for the connection. The connection has been established for 60 seconds, or 6000 timeticks.

Table 2-10 TCP Connection Table

ItcpConnTable	Elapsed	InBytes	InPkts	OutBytes	OutPkts
TCP A	6000	100	85	100	85
TCP B	4500	200	90	130	100
TCP C	9000	300	100	250	95

loctepConnElapsed

Provides the length of time that the TCP connection has been established.

Syntax: Timeticks

Access: Read-only

loctepConnInBytes

Provides the number of input bytes for the TCP connection.

Syntax: Counter

Access: Read-only

Transmission Control Protocol (TCP) Group

loctcpConnInPkts

Provides the number of input packets for the TCP connection.

Syntax: Counter

Access: Read-only

loctcpConnOutBytes

Provides the number of output bytes for the TCP connection.

Syntax: Counter

Access: Read-only

loctcpConnOutPkts

Provides the number of output packets for the TCP connection.

Syntax: Counter

Access: Read-only

End of Table