CHAPTER 5

In-Band Management

This chapter describes how to configure and manage the Catalyst 1700 using standard SNMP-based network management systems. This chapter also describes the standard MIBs and MIB extensions supported by the Catalyst 1700 and includes information on pre-compiled Schemas for SunNet Manager and Profiles for Novell NetWare Management System (NMS).

Management control of the Catalyst 1700 consists of the ability to monitor and configure the Catalyst 1700 and each of its ports. This in-band management is accomplished using a third party network management platform and application. The application communicates with the Catalyst 1700 in-band management interface (the SNMP agent) using the industry standard Simple Network Management Protocol.

This management information is represented as objects in a database called a MIB (Management Information Base). The Catalyst 1700 supports standard MIB II objects as well as custom extensions designed to maximize the manageability of the Catalyst 1700's switching and hub capabilities. These MIBs are described in this chapter.

Catalyst 1700 and SNMP Management Platforms

Using an SNMP management platform, you can locate/position a generic device icon for the Catalyst 1700 on the platform's network topology map. (See your network management system documentation for details.)

After selecting the Catalyst 1700, you can access a table of Catalyst 1700 objects that allow you to view the characteristics and counters describing the Catalyst 1700 and set certain object values as defined in the Catalyst 1700 MIB.

In order to gain these capabilities, you must configure the SNMP management platform to understand and be able to access the objects contained in the Catalyst 1700 MIB.

For SunNet Manager, you can use the supplied Schemas which specify the Catalyst 1700 MIB with the syntax specific to that particular management platform. For Novell NMS, you can use the supplied Profiles. The instructions for both of these are described later in this chapter. For other SNMP management systems, refer to the vendor's documentation for compiling third-party MIBs for use by a MIB Browser function.

Note Before you begin, the Catalyst 1700 must be configured for SNMP management. To do this, you must first assign an IP address to the Catalyst 1700 using the Internetwork Connection menu of the Out-of-Band Management system or using the Bootstrap protocol (BOOTP) discussed later in this chapter.

Supported TCP/IP Protocols

The Catalyst 1700 uses a subset of the Transmission Control Protocol/Internet Protocol (TCP/IP) suite as the underlying mechanism to transport the SNMP. The protocols implemented in the Catalyst 1700 follow:

- Internet Protocol (IP)
- Internet Control Message Protocol (ICMP)
- User Datagram Protocol (UDP)
- Trivial File Transfer Protocol (TFTP)
- Bootstrap Protocol (BOOTP)
- Address Resolution Protocol (ARP)

Configuring Catalyst 1700 for SNMP Management via BOOTP

The Catalyst 1700 must be configured with an IP address before it can make available any in-band management. You can assign addresses individually to each Catalyst 1700 in your administrative domain, or use the BOOTP to maintain a centralized database of such addresses.

To use BOOTP, you need a host machine with a BOOTP server software program. A database containing a list of physical MAC addresses and corresponding IP addresses must be set up on this host.

Other information such as the corresponding subnet masks, default gateway addresses and host names, may also be provided in the database, though these are strictly optional. The BOOTP server machine must be reachable by the Catalyst 1700 through one of its ports.

Upon a system reset, the Catalyst 1700 looks into its Non-Volatile Random Access Memory (NVRAM) for a configured IP address, and a default gateway address and IP subnet mask, if they exist.

If an IP address has not been configured, the Catalyst 1700 transmits a BOOTP broadcast request to all of its ports having a physical connection, requesting a mapping for its physical MAC address.

A valid response will include the IP address (mandatory), along with the subnet mask, the default gateway and the host name (all optional).

The reception of a valid BOOTP response immediately activates the rest of the hub's protocol suite, without requiring a system reset. The information is also saved in the NVRAM so the next reset will not have to redeploy BOOTP.

As long as its IP address remains undiscovered, the Catalyst 1700 will resend BOOTP requests indefinitely.

For more information about using BOOTP and adding the Catalyst 1700 to the address table, refer to your BOOTP server documentation.

Configuring Catalyst 1700 for SNMP Management via the Management Console and MIB Objects

You can also configure the IP address and related addresses using a combination of the out-of-band Management Console and MIB object extensions.

The first step is to configure the IP address using the Internetwork Connection menu of the Management Console (refer to “Out-of-Band Management”).

You can then either configure the corresponding subnet masks and default gateway addresses via the Management Consoles Internetwork Connection Configuration Menu or via the MIB objects described below.

Following each MIB object in parentheses, the type of value is shown. A brief description of the MIB object follows, as well as a list of valid values and a default value (if any).

netMgmtIpSubnetMask (IP Address)

This MIB reads and writes the Catalyst 1700 administrative IP subnet mask. The Catalyst 1700 may automatically discover a value for this object using the BootStrap protocol (BOOTP). The object value is also duplicated in the MIB II *ipAddrTable*. Once a value has been set for this object, the next write will take effect only after a unit reset.

Default Value: 0.0.0.0, or no subnet mask

netMgmtDefaultGateway (IP Address)

This MIB object reads and writes the IP address of a default gateway. A write to this object will take effect immediately, replacing the previous address, if any.

Default Value: 0.0.0.0, or no address

netMgmtIpAddress (IP Address)

This MIB object reads and writes the Catalyst 1700 administrative IP address. The Catalyst 1700 may automatically discover a value for this object using the BootStrap protocol (BOOTP). The object value is also duplicated in the MIB II *ipAddrTable*. Once a value has been set for this object, the next write will take effect only after a reset.

Default Value: 0.0.0.0, or no address

Community Strings

The Catalyst 1700 supports trivial authentication using community strings. The hub implements 2 separate community strings.

The first community string, known as the Get community string, has a default ASCII value of public, and can be used by a Network Management system to send Get and Get-Next requests to the Catalyst 1700.

The second string, the Set community string, has a default ASCII value of private, and may be used in Get, Get-Next and Set requests.

These strings are modifiable only through the out-of-band Management Console using the Network Management (SNMP) Configuration menu. (Refer to “Out-of-Band Management.”)

If configured to do so, the Catalyst 1700 will also generate authenticationFailure traps whenever it receives a request with an invalid community string as described later in this chapter.

Set Clients

To provide additional security, the Catalyst 1700 implements Set Client IP addresses. Up to 4 such IP addresses may be configured, denoting the list of management workstations allowed to issue Set requests.

The Set Client IP address list is initially empty (value: zero). Any Set Client workstation may then set the first address.

From this point on, only workstations having the same IP addresses as those on the list can add more Set Client IP addresses (the maximum total is 4), as well as setting other MIB objects. Otherwise, the set will silently be dropped.

To configure Set Managers use the following MIB objects:

netMgmtSetClientIndex (integer)

This MIB object returns the identification of a Set Client entry.

Valid Values: 1 to 4

netMgmtSetClientAddr (IP Address)

This MIB object reads and writes the Set Client's IP address. The Set Client is assumed to be Internet UDP/IP based.

Trap Clients/Traps

A Trap Client is a management workstation configured to receive and process traps. The Catalyst 1700 supports up to 4 Trap Clients.

The Trap Client list is shipped empty from the factory. An empty Trap Client list disables the generation of all traps. Traps use their own community strings and receiver addresses. To configure a Trap Client, the following MIB objects are used:

netMgmtTrapClientIndex (integer)

This MIB object returns the identification of a Trap Client entry.

Valid Values: 1 to 4

netMgmtTrapClientAddr (IP Address)

This MIB object reads and writes the Trap Client address. The Trap Client is assumed to be Internet UDP/IP based.

netMgmtTrapClientComm (display string)

This MIB object reads and writes the community string used for traps sent to this Trap Client.

The Catalyst 1700 can generate 5 standard traps plus 2 enterprise specific traps. MIB objects can be set to suppress the generation of the authenticationFailure traps and the linkUp/linkDown traps. These are detailed below, along with a description of all 7 traps.

coldStart Trap

This trap is generated upon a power-on reset, or upon the completion of a firmware upgrade where the new firmware is immediately selected for execution.

warmStart Trap

A warmStart trap is the result of the setting the *sysConfigReset* MIB object, sysConfigDefaultReset MIB object, or the execution of the reset command from one of the out-of-band Management Console menus.

linkDown Trap

This trap is produced whenever a port transitions into a suspended or disabled state due to a secure address violation (address mismatch or duplication), network connection error (loss of Link Beat, jabber error) or an explicit management disable action.

The trap frame carries the index value of the affected port. The following MIB is used to enable or disable the generation of this trap.

netMgmtEnableLinkTraps (integer)

This MIB object indicates whether the Catalyst 1700 is permitted to generate linkUp/linkDown traps. The value of this object overrides any configuration information; as such, it provides a means whereby all linkUp/linkDown traps may be disabled.

Valid Values: enabled (1)

disabled (2)

Default Values: enabled (1)

linkUp Trap

This trap reports a port transition from a suspended or disabled state to the enabled state. The trap frame contains the index value of the affected port. The following MIB is used to enable or disable the generation of this trap:

netMgmtEnableLinkTraps (integer)

This MIB object indicates whether the Catalyst 1700 is permitted to generate linkUp/linkDown traps. The value of this object overrides any configuration information; as such, it provides a means whereby all linkUp/linkDown traps may be disabled.

Valid Values: enabled (1)

disabled (2)

Default Values: enabled (1)

authenticationFailure Trap

This trap is generated whenever the Catalyst 1700 receives an SNMP message that is not properly authenticated (does not carry a valid community string). The following MIB is used to set this trap:

netMgmtEnableAuthenTraps (integer)

This MIB object indicates whether the Catalyst 1700 is permitted to generate authenticationFailure traps. The value of this object overrides any configuration information; as such, it provides a means whereby all authenticationFailure traps may be disabled.

This MIB object manipulates the same value for the snmpEnableAuthenTraps object instance.

Valid Values: enabled (1)

disabled (2)

Default Value: enabled (1)

logonIntruder Trap

This enterprise specific trap is produced whenever the out-of-band Management Console experiences successive logon failures due to incorrect passwords. The logon attempt threshold is user configurable using the *netMgmtConsolePasswordThresh* MIB object. Depending on how the MIB object *netMgmtConsoleSilentTime* is configured, the Catalyst 1700 may shut down the Management Console following the generation of this trap. The trap frame contains the name of the sending Catalyst 1700 (the value of the sysName MIB object, if available), or a null name.

switchDiagnostic Trap

The Catalyst 1700 issues this enterprise specific trap when its Power On Self Test (POST) code does not pass all tests. Some POST failures are fatal and may prevent the generation of this trap. The trap frame contains the name of the sending Catalyst 1700 (the value of the *sysName* MIB object, if available) or a null name. A trap client may query the failing Catalyst 1700 for the actual failure codes as stored in the *sysInfoPOSTResult* and *sysInfoPOSTPortVector* MIB objects.

Standard MIBs and MIB Extensions

The Catalyst 1700 implements all groups in MIB II except the Transmission Control Protocol and the Exterior Gateway Protocol.

The Catalyst 1700 RS-232 port uses the MIB described in the proposed RS-232 MIB standard, RFC-1317. All but the Synchronous Port group apply to the use of the Catalyst 1700 and are supported. This MIB and its associated MIB objects are defined later in this chapter.

The Catalyst 1700 also implements extensions to the standard MIB II in the form of an enterprise-specific MIB. These extensions provide for the management of the Catalyst 1700 physical ports and its unique bridging and switching capabilities.

The Catalyst 1700 Personal Ethernet ports are numbered P1-P24, the General Ethernet port is G25, and the Fast Ethernet ports are F1 and F2. Index variables for these ports are 1 through 24 for the Personal Ethernet ports, 25 for the General Ethernet port, and 26 and 27 for Fast Ethernet ports F1 and F2 respectively.

The table below lists many of the actions related to managing and configuring the Catalyst 1700 and the MIB objects associated with each action. It includes Catalyst 1700 MIB extensions, as well as RFC-1317 MIB objects. Following the table are descriptions of each MIB object, its valid values and its default value.

Table 5-1 Catalyst 1700 MIB Objects

Action	Associated MIB Objects
View Self Test Results	sysInfoPOSTResult
	sysInfoPOSTPortVector
View System Information	sysInfoFwdEngineRevision
	sysInfoBoardRevision
	sysInfoTotalNumberOfPorts
	sysInfoNumberOfSwitchPorts
	sysInfoNumberOfSharedPorts
	swPortControllerRevision
View/Configure RS-232 Port Characteristics	232Number
	rs232PortIndex
	rs232PortType
	rs232PortInSigNumber
	rs232PortOutSigNumber
	rs232PortInSpeed
	rs232PortOutSpeed
View/Configure RS-232 Async Port Characteristics	rs232AsyncPortIndex
	rs232AsyncPortBits
	rs232AsyncPortStopBits
	rs232AsyncPortParity
	rs232AsyncPortAutobaud

Action	Associated MIB Objects
View RS-232 Async Port Statistics	rs232AsyncPortParityErrs
	rs232AsyncPortFramingErrs
	rs232AsyncPortOverrunErrs
ActionAssociated MIB ObjectsView RS-232 Port Input/Output Signals	rs232InSigPortIndex
	rs232InSigName
	rs232InSigState
	rs232InSigChanges
	rs232OutSigPortIndex
	rs232OutSigName
	rs232OutSigState
	rs232OutSigChanges
View/Configure RS-232 Port for an Attached Modem	netMgmtAutobaud
	netMgmtModemInitString
	netMgmtModemAutoAnswer
	netMgmtModemDialString
	netMgmtModemDialDelay
View/Configure Logon Security	netMgmtConsolePasswordThresh
	netMgmtConsoleSilentTime
	netMgmtConsoleInactTime
View/Configure Learning Mode	swPortMatchAddress
View/Configure Switching Mode	sysConfigSwitchingMode
View/Configure Port Monitoring Mode	sysConfigMonitor
	sysConfigMonitorPort
	sysConfigHigherProtMonitor
	swPortMonitoring

Action	Associated MIB Objects
View/Configure Inactivity Aging Periods	sysConfigLbLossAgeTime
	sysConfigInactAgeTime
	netMgmtConsoleInactTime
View/Configure Performance Information	sysInfoBuffersUsed
	sysInfoMaxBuffers
	sysInfo10MbpsBandwidthExceeds
	sysInfoCurrentBandwidth
	sysInfoPeakBandwidth
	sysInfoPeakTime
	sysInfoUtilDisplay
	sysConfig10MbpsBandwidthExceeds
	sysConfigPeakBandwidth
	sysInfoQueueFullOccured
View/Configure Address Security	swPortMatchAddress
	sysConfigAddressViolationAction
	swPortSecureAddressDuplication
	swPortSecureAddressMismatches
	swPortLastSourceAddress
View/Configure Port Characteristics	swPortIndex
	swPortDescr
	swPortType
	swPortMtu
	swPortSpeed
	swPortConnectorType

Action	Associated MIB Objects
View/Configure Port Status	swPortStatus
	swPortAdminStatus
	swPortLastStatus
	swPortStatusChanges
	swPortLinkbeatStatus
	swPortLinkbeatLosses
	swPortJabberStatus
	swPortJabbers
View/Configure Port Address Status	swPortAddressChanges
	swPortAddressStatus
	swPortAdminAddressStatus
	swPortLastSourceAddress
	sysConfigLbLossAgeTime
	sysConfigInactAgeTime
View/Configure Network Port	sysInfoNetPort
	sysConfigNetPortAdmin
	sysConfigG25Connector
	swPortConnectorType

Action	Associated MIB Objects
View Port Receive Statistics	swPortRxStatIndex
	swPortRxOctets
	swPortRxOctetsWraps
	swPortRxFrames
	swPortRxGroupFrames
	swPortRxForwardedFrames
	swPortRxFilteredFrames
	swPortRxNoBufferDiscards
	swPortRxFCSErrors
	swPortRxAlignmentErrors
	swPortRxFrameTooLongs
	swPortRxRunts
View Port Transmit Statistics	swPortTxStatIndex
	swPortTxOctets
	swPortTxOctetsWraps
	swPortTxFrames
	swPortTxDeferrals
	swPortTxSingleCollisions
	swPortTxMultipleCollisions
	swPortTxLateCollisions
	swPortTxExcessiveCollisions
	swPortTxExcessiveDeferrals
	swPortTxExcessiveCollisions16s
	swPortTxErrors
View/Configure Collision Histograms	swPortTxCollIndex
	swPortTxCollCount
	swPortTxCollFrequencies

Action	Associated MIB Objects
View/Configure for SNMP In-Band Management	netMgmtIpAddress
	netMgmtDefaultGateway
	netMgmtIpSubnetMask
View/Configure Set Clients	netMgmtSetClientIndex
	netMgmtSetClientAddr
View/Configure Trap Clients and Traps	netMgmtTrapClientIndex
	netMgmtTrapClientAddr
	netMgmtTrapClientComm
	netMgmtEnableLinkTraps
	netMgmtEnableAuthenTraps
	logonIntruder
	switchDiagnostic
View/Configure Firmware Upgrades	upgradeFirmwareSource
	upgradeEPROMRevision
	upgradeFlashSize
	upgradeFlashBankStatus
	upgradeTFTPServerAddress
	upgradeTFTPLoadFilename
	upgradeTFTPInitiate
	upgradeAutoExecute
	upgradeTFTPAccept
Reset System	sysConfigReset
	sysConfigDefaultReset
Clear Port Statistics	sysConfigClearPortStats

RFC-1317 MIB Object Listing

The Catalyst 1700 supports the following RFC-1317 groups: the Generic RS-232-Like Group, the RS-232-Like General Port Table, the RS-232-Like Asynchronous Port Group, the Input Signal Table, and the Output Signal Table. The associated MIB objects for these groups are outlined below.

Following each MIB object in parentheses, the type of value is shown. A brief description of the MIB object follows, as well as a list of valid values and a default value (if any).

Generic RS-232-Like Group

rs232Number (integer)

This read-only MIB object returns the number of ports (regardless of their current state) in the RS-232-Like General Port Table.

Valid Value: 1

rs232PortIndex (integer)

A unique value for each port is contained in this read-only MIB. Its value ranges between 1 and the value of *rs232Number.*

rs232PortType (integer)

The port's hardware type is returned with this MIB object.

Valid Value: rs232 (2)

rs232PortInSigNumber (integer)

This MIB object reads the number of input signals for the port in the Input Signal Table (*rs232PortInSigTable*).

rs232PortOutSigNumber (integer)

This MIB object contains the number of output signals for the port in the Output Signal Table (*rs232PortOutSigTable*).

rs232PortInSpeed (integer)

This MIB object reads and writes the port's input speed in bits per second.

Default Value: 9600

rs232PortOutSpeed (integer)

The port's output speed in bits per second can be read or set in this MIB object.

Default Value: 9600

RS-232 Like Asynchronous Port Group

rs232AsyncPortIndex (integer)

A unique value for each port is returned with this MIB object. Its value is the same as *rs232PortIndex* for the port.

rs232AsyncPortBits (integer)

This MIB object reads and writes the port's number of bits in a character.

Default Value: 8

rs232AsyncPortStopBits (integer)

This MIB object reads and writes the port's number of stop bits.

Valid Values:	one	(1)
	two	(2)
	one-and-half	(3)
	dynamic	(4)
Default Value:	one	(1)

rs232AsyncPortParity (integer)

The port’s sense of a character parity bit can be read and set with this MIB object.

Valid Values:	none	(1)
	odd	(2)
	even	(3)
	mark	(4)
	space	(5)
Default Value:	none	(1)

rs232AsyncPortAutobaud (integer)

A control for the port’s ability to automatically sense input speed is read and set with this MIB object.

When *rs232PortAutoBaud* is enabled, a port may autobaud to values different from the set value for speed. As a result, a network management system may temporarily observe a value different from the previously set value.

Valid Values: enabled (1)

disabled (2)

Default Value: enabled (1)

rs232AsyncPortParityErrs (counter)

This MIB object returns the total number of characters with a parity error, input from the port since system re-initialization and while the port state was up or test.

rs232AsyncPortFramingErrs (counter)

The total number of characters with a framing error, input from the port since system re-initialization and while the port state was up or test is returned with this MIB object.

rs232AsyncPortOverrunErrs (counter)

This MIB object contains the total number of characters with an overrun error, input from the port since system re-initialization and while the port state was up or test.

Input Signal Group

rs232InSigPortIndex (integer)

The value of *rs232PortIndex* for the port to which this entry belongs is contained in this MIB object.

rs232InSigName (integer)

This MIB object reads the identification of a hardware signal.

Valid Values:	rts	(1):	Request to Send
	cts	(2):	Clear to Send
	dsr	(3):	Data Set Ready
	dtr	(4):	Data Terminal Ready
	ri	(5):	Ring Indicator
	dcd	(6):	Received Line Signal Detector
	sq	(7):	Signal Quality Detector
	srs	(8):	Data Signaling Rate Selector
	srts	(9):	Secondary Request to Send
	scts	(10):	Secondary Clear to Send
	sdcd	(11):	Secondary Received Line Signal Detector

rs232InSigState (integer)

This MIB object returns the current signal state.

Valid Values:	none	(1)
	on	(2)
	off	(3)

rs232InSigChanges (counter)

The number of times the signal has changed from on to off or from off to on is contained in this MIB object.

Output Signal Group

rs232OutSigPortIndex (integer)

This MIB object reads the value of *rs232PortIndex* for the port to which this entry belongs.

rs232OutSigName (integer)

This MIB object returns the identification of a hardware signal.

Valid Values:	rts	(1):	Request to Send
	cts	(2):	Clear to Send
	dsr	(3):	Data Set Ready
	dtr	(4):	Data Terminal Ready
	ri	(5):	Ring Indicator
	dcd	(6):	Received Line Signal Detector
	sq	(7):	Signal Quality Detector
	srs	(8):	Data Signaling Rate Selector
	srts	(9):	Secondary Request to Send
	scts	(10):	Secondary Clear to Send
	sdcd	(11):	Secondary Received Line Signal Detector

rs232OutSigState (integer)

The current signal state is contained in this MIB object.

Valid Values: none (1)

on (2)

off (3)

rs232OutSigChanges (counter)

This read-only MIB object returns the number of times the signal has changed from on to off or from off to on.

Catalyst 1700 MIB Extensions Object Listing

The Catalyst 1700 MIB extensions are grouped into 5 categories: System Information (sysInfo), System Configuration (sysConfig), Port (port), Network Management (netMgmt) and Upgrade (upgrade). These groups and their associated MIB objects are detailed in the following section.

System Information

All objects in this group are read-only and return system-wide information. This group is one set of extensions to the MIB II System Group, with some objects serving as mirror images of objects in the System Configuration (sysConfig) Group.

Following each MIB object in parentheses, the type of value is shown. A brief description of the MIB object follows, as well as a list of valid values and a default value (if any).

sysInfoFwdEngineRevision (integer)

This read-only MIB object returns the revision number of the forwarding engine controller.

sysInfoBoardRevision (integer)

This read-only MIB object returns the revision number of the main Catalyst 1700 board on which the Catalyst 1700 firmware resides.

sysInfoTotalNumberOfPorts (integer)

This read-only MIB object responds with the total number of Personal Ethernet ports, General Ethernet ports, and Fast Ethernet ports in this Catalyst 1700 unit.

Valid Value: 2
7

sysInfoBuffersUsed (gauge)

This read-only MIB object displays the high-water mark of frame buffer usage in this Catalyst 1700.

Default Value: zero

sysInfoMaxBuffers (counter)

This MIB object contains the number of times *sysBuffersUsed* has reached its maximum value and been reset to zero by the Catalyst 1700 firmware.

Default Value: zero

sysInfoPeakBandwidth (counter)

This MIB object shows the highest bandwidth utilized since system reset or last cleared. The measurement unit is in kilobits per second (1000 bits/second). Writing a zero to the companion object *sysConfigPeakBandwidth* clears this object.

sysInfoPeakTime (display string)

Up to 32 characters containing the date and time the value in *sysInfoPeakBandwidth* is captured and shown in this MIB object. The string is in net ASCII and conforms exactly to the following format:

Mon Sep 21 07:02:01 1995

sysInfoCurrentBandwidth (counter)

This MIB object shows the bandwidth currently consumed. The measurement unit is in kilobits per second (1000 bits/second).

sysInfo10MbpsBandwidthExceeds (counter)

This MIB object contains a count of the number of times bandwidth consumption exceeded 10 megabits per second (the maximum bandwidth of a 10Base-T hub).

sysInfoNetPort (integer)

This MIB object responds with the number of the port currently operating as the Network port. This number is either learned automatically, or assigned by the user in a companion writable object.

Valid Values:	self-sensing	(1)
	g25	(2)
	f1	(3)
	f2	(4)
Default Value:	self-sensing	(1)

sysInfoUtilDisplay (integer)

The number of utilization meter LEDs currently lit on the front panel is contained in this read-only MIB object.

sysInfoQueueFullOccured (integer)

This read-only MIB object indicates whether the Catalyst 1700 has had to discard some packets because a port transmit queue is full.

Valid Values: true (1)

false (2)

sysInfoPOSTResult (integer)

This read-only MIB object returns a bit array where the presence of a particular bit indicates a failure of a particular Power On Self Test (POST).

Valid Values:

Bit	POST Test
2^0	ports
2^1	station address PROM
2^2	watchdog interrupt
2^3	RS-232 port
2^4	real time clock
2^5	system timer interrupt
2^6	port control/status
2^7	DRAM
2^8	DRAM
2^9	forwarding engine
2^10	forwarding engine
2^11	non volatile RAM
2^12	SRAM
2^13	EPROM

sysInfoPOSTPortVector (integer)

This read-only MIB object displays a bit array where the presence of a particular bit indicates a failure of a particular port, as determined by the POST individual ports test.

Valid Values:

Bit	Failed port
2^0	P1
2^1	P2
2^2	P3
2^3	P4
2^4	P5
2^5	P6
2^6	P7
2^7	P8
2^8	P9
2^9	P10
2^10	P11
2^11	P12
2^12	P13
2^13	P14
2^14	P15
2^15	P16
2^16	P17

Valid Values:

Bit	Failed port
2^17	P18
2^18	P19
2^19	P20
2^20	P21
2^21	P22
2^22	P23
2^23	P24
2^24	G25
2^25	F1
2^26	F2

sysInfoNumberOfSwitchPorts (integer)

This read-only MIB object returns the number of ports that are individually switched. Each of these ports has hardware support for a full range of statistics and management control. This object represents the upper bound of indices into the various switch port tables in the port group.

Valid Values: 2
7

System Configuration

This group contains another set of extensions to the MIB II Systems Group. All objects in this group are read-write, with some objects serving as writable mirror images of objects in the System Information (sysInfo) Group.

sysConfigReset (integer)

Setting this object to reset causes a complete reset of both hardware and software, but does not run the POST (Power On Self Test). Setting this object to noReset has no effect. The Catalyst 1700 always returns the value noReset when this object is read.

Valid Values: noReset (1)

reset (2)

Default Value: noReset (1)

sysConfigDefaultReset (integer)

Setting this object to reset causes a complete reset of both hardware and software, but does not run the POST (Power On Self Test). All configuration parameters will revert to their factory default settings, and all addresses assigned or learned will be removed. The content of FLASH memory remains unchanged. The Catalyst 1700 will execute the firmware from the same source (EPROM or FLASH) after coming out of reset.

Setting this object to noReset has no effect. The Catalyst 1700 always returns the value noReset when this object is read.

Valid Values: noReset (1)

reset (2)

Default Value: noReset (1)

sysConfigPeakBandwidth (integer)

Writing a zero to this MIB object clears the companion object *sysInfoPeakBandwidth*.

sysConfig10MbpsBandwidthExceeds (integer)

Writing a zero to this MIB object clears the companion object *sysInfo10MbpsBandwidthExceeds*.

sysConfigAddressViolationAction (integer)

Setting this MIB object indicates what action to take whenever a Secure Address Violation (an address mismatch or duplication) occurs.

Valid Values: suspend (1)

disable (2)

noAction (3)

Default Value: suspend (1)

sysConfigLbLossAgeTime (integer)

This MIB object reads and sets the number of seconds to wait after a Link Beat is lost, before making a non-network dynamic port Unaddressed. If a value of 0 is specified, a port without Link Beat will not be transitioned to Unaddressed. This aging interval is global and applies to all ports.

Valid Values: 0 to 65500

Default Value: 0

sysConfigInactAgeTime (integer)

This MIB object reads and sets the number of seconds of inactivity (no received frames) to wait before making a non-network dynamic port Unaddressed. If a value of 0 is specified, The Catalyst 1700 will not make an inactive port Unaddressed. This aging interval is global and applies to all ports.

Valid Values: 0 to 65500

Default Value: 0

Note Incorrect setting of address aging intervals may cause a client session to time-out.

When configuring a Catalyst 1700 to age a port address, be sure that the address aging interval is set to higher time periods that the server/client poll session (commonly known as keep-alives).

If you are unsure of the keep-alive intervals of the server/client, the recommendation is to leave the aging intervals at the default settings of None (no aging).

sysConfigSwitchingMode (integer)

The Catalyst 1700 can operate as a high speed cut-through switch, or as a traditional store-and-forward bridge. Cut-through switching is enabled when this MIB object is set to FastForward or FragmentFree. This works to reduce buffering latency by attempting to forward a frame before reception is completed. Cut-through switching does not take effect for frames that are addressed to a multicast or broadcast address, or when a monitor port is configured.

Valid Values:	store-and-Forward	(1)
	fragmentFree	(2)
	fastForward	(3)
Default Value:	fastForward	(3)

sysConfigMonitor (integer)

This MIB object selects whether frames to/from certain ports are sent to *sysConfigMonitorPort*. A port is selected for monitoring purpose when its *swPortMonitoring* object is set to enabled.

Valid Values: enabled (1)

disabled (2)

Default Value: disabled (2)

sysConfigMonitorPort (integer)

This MIB object responds with the port number to which all frames to/from monitored ports are sent. Port 0 (zero) indicates that the frames are to be monitored internally by the Catalyst 1700. Frames are only monitored if the *sysConfigMonitor* object is set to enabled.

Valid Values: 0 to *sysInfoTotalNumberOfPorts*

Default Value: 0

sysConfigHigherPortMonitor (integer)

This MIB object selects whether frames addressed to and from the higher layer protocol processor of the Catalyst 1700 are to be monitored. The higher layer protocol processor is the entity that responds to all SNMP/TFTP management requests.

Valid Values: enabled (1)

disabled (2)

Default Value: disabled (2)

sysConfigNetPortAdmin (integer)

With this object, the user can explicitly specify which port is to be the network port.

Valid Values:	self-sensing	(1)
	g25	(2)
	f1	(3)
	f2	(4)
Default Value:	self-sensing	(1)

sysConfigG25Connector (integer)

This MIB object specifies which of the three G25 connector types is providing the active connection. Setting the connector to self-sensing allows the Catalyst 1700 to learn the active connector for G25 on its own. The other 3 possible values force the Catalyst 1700 to use the chosen type.

Valid Values:	self-sensing	(1)
	rj45	(2)
	bnc	(3)
	aui	(4)
Default Value:	self-sensing	(1)

sysConfigClearPortStats

Setting this MIB object to noClear results in no action. When set to clear, the Catalyst 1700 will reset all swPort statistics to zero. Port statistics are kept in the *swPortTable*, the *swPortTxStatTable*, the *swPortRxStatTable* and the *swPortCollTable*. This object always returns noClear when read.

Valid Values: clear (1)

noClear (2)

Switch Port

This group of extensions provides for the configuration and management of individual Catalyst 1700 ports.

Port Configurations

This group of extensions provide general configurations, some of which are read-only, some of which are read/write, for a corresponding port.

swPortIndex (integer)

This MIB object responds with a number from 1 to *sysInfoNumberOfSwitchPorts* identifying a particular port. The same value of a port index variable for any of the port tables selects the same port.

swPortControllerRevision (integer)

This MIB object returns the revision of the port controller.

swPortDescr (display string)

This read/write MIB object contains a descriptive string of up to 60 characters used by the network administrator to name a port.

swPortType (integer)

This MIB object responds with the port type. Port types include Personal-Ethernet (dedicated 10 Mbps Ethernet ports), general-Ethernet (dedicated or shared 10 Mbps Ethernet ports), general-fast-Ethernet (dedicated or shared 100 Mbps Ethernet ports), personal-fast-Ethernet (dedicated 100Mbps Ethernet ports) or other (none of the preceding).

Valid Values:	other	(1)
	personal-ethernet	(2)
	general-ethernet	(3)
	general-fast-ethernet	(4)
	personal-fast-ethernet	(5)

swPortMtu (integer)

This MIB object displays the port maximum transmission unit size; it is always 1518 for the Catalyst 1700.

swPortSpeed (integer)

This MIB object displays the port speed. The value is always 10,000,000 bps if *swPortType* is personal-Ethernet or general-Ethernet. It is 100,000,000 bps, otherwise.

swPortConnectorType (integer)

This read-only MIB object responds with the type of connector the port is currently using.

Valid Values:

other	(1)
rj45	(2)
bnc	(3)
aui	(4)
fast-enet	(5)

swPortStatus (integer)

This read-only MIB object displays the current operational status of the port. An *swPortStatus* of disabled-mgmt, disabled-mismatch or disabled-duplication is saved across a reset, so a port may also come up with such a status.

Valid Values:

enabled	(1):	normal operation (transmit and receive)
disabled-mgmt	(2):	disabled by explicit management action
suspended-linkbeat	(3):	suspended due to absence of Link Beat
suspended-jabber	(4):	suspended because port is jabbering
suspended-duplication	(5):	suspended due to a duplicate address
suspended-mismatch	(6):	suspended due to an address mismatch
disabled-duplication	(7):	disabled due to a duplicate address
disabled-mismatch	(8):	disabled due to an address mismatch

Default Value:

enabled	(1):	unless POST has detected a failure on the port, in which case, it is disabled-mgmt (2)

swPortAdminStatus (integer)

By explicit management action, this MIB object changes the port status to either enabled: normal status (transmit and receive) or disabled-mgmt: transmit and receive disabled.

Valid Values:	enabled	(1)
	disabled-mgmt	(2)
Default Value:	enabled	(1)

swPortLastStatus (integer)

This MIB object returns the value of *swPortStatus* prior to its current value, or enabled, if *swPortStatusChanges* is 0.

Valid Values:	enabled	(1)
	disabled-mgmt	(2)
	suspended-linkbeat	(3)
	suspended-jabber	(4)
	suspended-duplication	(5)
	suspended-mismatch	(6)
	disabled-duplication	(7)
	disabled-mismatch	(8)
Default Value:	enabled	(1)

swPortStatusChanges (counter)

This MIB object responds with the number of times swPortStatus has changed.

swPortAddressStatus (integer)

This MIB objects reads and writes the current address status of the port. Ports can be Unaddressed: No address has been learned or assigned; Single: One address has been learned or assigned; or Network: A network port which can receive with any address. Only one port in the Catalyst 1700 can have the Network address status.

Valid Values:	unaddressed	(1)
	single	(2)
	network	(3)
Default Value:	unaddressed	(1)

swPortAdminAddressStatus (integer)

By explicit management action, this MIB object changes the address status of a non-network port to Unaddressed: No address has been learned or assigned. A read of this object may return one of two other possible address status values, namely Single and Network, as implicitly administered by the system, or by explicit management actions on related port objects.

Valid Values:	unaddressed	(1)
	single	(2)
	network	(3)
Default Value:	unaddressed	(1)

swPortMonitoring (integer)

This MIB object sets the Catalyst 1700 to copy the receive and transmit traffic on this port to *sysConfigMonitorPort*.

Valid Values: enabled (1)

disabled (2)

Default Value: disabled (2)

swPortMatchAddress (address)

This MIB object sets a port to receive a particular address. This action enables or disables Secure Learning Mode for the corresponding port.

Valid Values: set to zeroes to unsecure the address (Dynamic learning). Set to all ones (hexadecimal ffffffffffff) to secure the next source address received. Entry of a valid address secures that address.

Default Value: zero

swPortSecureAddressDuplications (counter)

This MIB object displays the number of times a source address was seen on this port which duplicated a secure address configured on another port.

swPortSecureAddressMismatches (counter)

This MIB object returns the number of times a source address was seen on this port which differed from the secure address configured for the port.

swPortLastSourceAddress (IP Address)

This MIB object contains the last source address seen by the Catalyst 1700 in a frame received from this port.

swPortLinkbeatStatus (integer)

This MIB object displays the current port Link Beat status.

Valid Values: linkbeat (1)

noLinkbeat (2)

swPortLinkbeatLosses (counter)

This MIB object responds with the number of times that the value of *swPortLinkbeatStatus* has changed to *noLinkbeat*.

swPortJabberStatus (integer)

This MIB object displays the current port jabber status.

Valid Values: notJabbering (1)

jabbering (2)

swPortJabbers (counter)

This MIB object contains the number of times that the jabber function has to be invoked because a frame transmitted from this port exceeded a certain time duration.

swPortAddressChanges (counter)

This MIB object contains the number of times the address on this port has changed as detected by the Catalyst 1700 firmware. This counter is only updated when the port is NOT the network port.

An address change occurs when the port learns or is assigned a new address.

Port Receive Statistics

These statistics are related to reception activities on the ports.

swPortRxStatIndex (integer)

This MIB object reads and writes a number from 1 to *sysInfoNumberOfSwitchPorts* identifying a particular port. The same value of a port index variable for any of the port tables selects the same port.

swPortRxOctets (counter)

This MIB object returns a count of data and padding octets in frames that are successfully received. This does not include octets in frames received with frame-too-long, FCS, length or alignment errors, or frames lost due to an internal MAC sublayer error.

swPortRxOctetsWraps (counter)

This MIB object contains the number of times swPortRxOctets has rolled to zero.

swPortRxFrames (counter)

This MIB object displays the count of frames that are successfully received. This does not include frames received with frame-too-long, FCS, length or alignment errors, or frames lost due to an internal MAC sublayer error.

swPortRxGroupFrames (counter)

This MIB object returns a count of frames that are successfully received and are directed to a group (broadcast or multicast) address. This does not include frames received with frame-too-long, FCS, length or alignment errors, or frames lost due to an internal MAC sublayer error.

swPortRxForwardedFrames (counter)

The count of frames received and forwarded are contained in this MIB object.

swPortRxFilteredFrames (counter)

A count of frames received that were discarded due to a failure of a Destination Address comparison is contained in this MIB object. Two scenarios can occur which will increment this counter. The first occurs if no Network port exists and a Destination Address comparison fails. The second occurs if a frame is received through the Network port and a Destination Address comparison fails.

swPortRxNoBufferDiscards (counter)

This MIB object returns a count of frames received that were discarded due to a lack of frame buffer resources in the Catalyst 1700.

swPortRxFCSErrors (counter)

This MIB object displays a count of frames received that are an integral number of octets in length but do not pass the Frame Check Sequence test.

swPortRxAlignmentErrors (counter)

A count of frames received that are not an integral number of octets in length and do not pass the Frame Check Sequence test is contained in this MIB object.

swPortRxFrameTooLongs (counter)

This MIB object returns a count of frames received that exceed the maximum permitted frame size.

swPortRxRunts (counter)

A count of frames received that are shorter than the minimum permitted frame size is returned with this MIB object. Runts usually indicate collision fragments, a normal network event.

Note The Catalyst 1700 will incorrectly record runts on the high speed (100 Mbps) ports that have not actually been received. Therefore, the runt statistics for the high speed ports are not meaningful and should be ignored.

Port Transmit Statistics

These statistics are related to transmit activities on the ports.

swPortTxStatIndex (integer)

This MIB object reads and writes a number from 1 to *sysInfoNumberOfSwitchPorts* identifying a particular port. The same value of a port index variable for any of the port table selects the same port.

swPortTxOctets (counter)

This MIB object contains a count of data and padding octets of frames that are successfully transmitted.

swPortTxOctetsWraps (counter)

The number of times *swPortTxOctets* has rolled to zero is contained in this MIB object.

swPortTxFrames (counter)

This MIB object returns a count of frames that are successfully transmitted.

swPortTxDeferrals (counter)

This MIB object displays a count of frames for which the first transmission attempt is delayed because the medium is busy.

swPortTxSingleCollisions (counter)

A count of successfully transmitted frames for which transmission is inhibited by exactly one collision is contained in this object.

swPortTxMultipleCollisions (counter)

This MIB object returns a count of successfully transmitted frames for which transmission is inhibited by more than one collision.

swPortTxLateCollisions (counter)

The number of times that a collision is detected later than 512 bit-times into the transmission of a frame is contained in this MIB object. A late collision is also considered as a (generic) collision for purposes of other collision-related statistics.

swPortTxExcessiveCollisions (counter)

This MIB object returns a count of frames for which transmission fails due to excessive collisions (16).

swPortTxExcessiveDeferrals (counter)

A count of frames for which transmission is deferred for an excessive period of time is contained in this MIB object.

swPortTxCollision16s (counter)

This MIB object returns a count of frames that due to excessive collisions are not transmitted successfully. Here, the normal threshold of 16 collisions is used to determine the status of the transmission. A *swPortExcessiveCollision16s* is counted twice, i.e., both as a *swPortExcessiveCollision16s* and as a *swPortExcessiveCollisions*.

swPortTxErrors (counter)

This MIB object displays a count of frames for which transmission failed due to an internal MAC sublayer transmit error. A frame is only counted by an instance of this object if it is not counted by the corresponding instance of either the *swPortLateCollisions* object, the *swPortExcessiveCollisions* object, the *swPortExcessiveCollision16s* object, or the *swPortExcessiveDeferrals* object.

Collision Histogram

This group contains histograms related to collisions experienced by each of the ports.

swPortTxCollIndex (integer)

This MIB object contains a number from 1 to *sysInfoNumberOfSwitchPorts* identifying a particular physical port. The same value of a port index variable for any of the port tables selects the same port.

swPortTxCollCount (integer)

This MIB object contains the number of per-frame media collisions for which a particular collision histogram cell represents the frequency on a particular port.

swPortTxCollFrequencies (counter)

This MIB object contains a count of frames for which the transmission (successful or otherwise) on a particular port is accompanied by a particular number of media collisions.

Management Console

The following MIB objects relate to the Catalyst 1700 Management Console described in "Out-of-Band Management." They can be used to configure modem communication parameters, as well as security features. To configure the physical characteristics of the RS-232 port, the proposed RS-232 standard MIB, RFC-1317, should be used. The RFC-1317 MIB is detailed earlier in this chapter.

netMgmtModemInitString; (display string)

This MIB object reads and writes the initialization string used to configure an attached modem. Specify this string if the modem is not Hayes compatible. Only the initialization commands need to be specified (i.e., do NOT specify any modem prefix string such as the AT attention sequence, or any suffix such as the CR character). This string takes effect after every line hangup.

Default Value: E0V1M1

netMgmtModemDialString (display string)

This read/write MIB object displays the string containing a phone number which is used to establish a modem connection with a remote site.

Only the phone number needs to be specified (i.e., do NOT specify any dial prefix string such as ATDT). Leave this string empty if this dial-out capability is not desired. This string takes effect after every line hangup.

netMgmtModemDialDelay (integer)

This MIB object specifies a delay in seconds between every dial-out failure. The value zero indicates an infinite delay. Normally, the Catalyst 1700 will attempt a dial-out if the *netMgmtModemDialString* object is non-empty.

If the remote site then fails to answer, and auto-answer has been disabled on this system, the system will retry the dial-out attempt after this delay has passed. If auto-answer is enabled and the first dial-out attempt fails, the system will stop further dial-out attempts and immediately go into auto-answer mode.

Valid Values: 0 to 65500

Default Value: 300

netMgmtModemAutoAnswer (integer)

This MIB object specifies whether the system should be in auto-answer mode and only accept incoming calls. Note that the system will always attempt a dial-out first if the *netMgmtModemDialString* object is non-empty.

Valid Values: enabled (1)

disabled (2)

Default Value: enabled (1)

netMgmtConsoleInactTime (integer)

This MIB object reads and writes the number of seconds of Management Console session inactivity to wait before ending the session. Once a session has ended, the user must resupply the Management Console password to regain access. A value of 0 indicates no time-out.

Valid Values: 0 to 65500

Default Value: 3

netMgmtConsolePasswordThresh (integer)

This MIB object reads and writes the number of consecutive invalid password attempts allowed before the Management Console is shut down (kept silent) for a configured duration. A zero value permits unlimited attempts.

Valid Values: 0 to 65500

Default Value: 0

netMgmtConsoleSilentTime; (integer)

This MIB object reads and writes the number of minutes during which the Management Console will be unavailable after repeated failed attempts to logon. A zero value specifies no silent time.

Valid Values: 0 to 65500

Default Value: 0

Upgrade Group

This group contains a collection of Upgrade facility MIB objects. They are defined as follows:

upgradeFirmwareSource (integer)

This MIB object selects the source from which firmware is read. A valid selection will cause a reset, followed by the execution of the selected firmware. A valid selection is defined as:

- A selection which yields a firmware that is different from the one currently executing.
- If the selection is flash, Flash memory is available in the system and contains a checksum verified firmware.

Valid Values: eprom (1)

flash (2)

Default Value: eprom (1)

upgradeEPROMRevision (display string)

This MIB object returns the revision of the Catalyst 1700 firmware residing in EPROM. The string has the following format V1.0a.

upgradeFlashSize (integer)

This MIB object contains the size of available Flash memory in the system, in kilobytes. For example: 128 equals 131072 bytes (128 bytes times 1024).

upgradeFlashBankStatus (display string)

This MIB object displays a text string indicating status and version of last upgrade to Flash memory, including the source of the upgrade, version and the date and time of upgrade. The string is in net ASCII and conforms exactly to one of the following format, depending upon the source of the upgrade:

V1.0a written on Mon Sep 21 07:02:01 1992 from serial terminal: valid

or

V1.0a written on Tue Dec 01 15:19:15 1992 from 192.009.200.200: invalid

upgradeTFTPServerAddress (IP Address)

This MIB object reads and writes the IP address of a TFTP server from which a firmware image can be downloaded. The download may be initiated by setting the *upgradeTFTPInitiate* to upgrade or via an out-of-band management action.

Default Value: 0.0.0.0, or no address

upgradeTFTPLoadFilename (display string)

This MIB object reads and writes the name of the file containing a firmware upgrade image on the host whose address is given by the *upgradeTFTPServerAddress*.

Default Value: empty string, or no filename

upgradeTFTPInitiate (integer)

Setting this MIB object to noUpgrade results in no action.

When set to upgrade, the Catalyst 1700 will attempt to download a firmware upgrade image from the server whose address is given by *upgradeTFTPServerAddress*.

The image is found in the *upgradeTFTPLoadFilename*. Both *upgradeTFTPServerAddress* and *upgradeTFTPLoadFilename* must be non-empty for the upgrade to proceed. This object always returns noUpgrade when read.

Valid Values: upgrade (1)

noUpgrade (2)

Default Value: noUpgrade (2)

upgradeAutoExecute (integer)

This MIB object indicates whether a newly upgraded firmware version should immediately be selected for execution.

When this object is disabled, the user must explicitly set the *upgradeFirmwareSource* object to select and run a particular firmware version after an upgrade.

When this object is enabled, following a successful firmware upgrade, the system will automatically switch to run the new firmware.

Valid Values: enabled (1)

disabled (2)

Default Value: enabled (1)

upgradeTFTPAccept (integer)

This object controls the second method of firmware upgrade using TFTP. The Catalyst 1700 has a UDP Listener on the TFTP server port, and can accept upgrade requests from any workstation with Internet Protocol TFTP software.

This object enables or disables the TFTP Upgrade Listener. When disabled, no TFTP workstations can download firmware upgrade images to the Catalyst 1700.

Valid Values: enabled (1)

disabled (2)

Default Value: enabled (1)

MIB Compilation in a Novell NMS Environment

This section describes how to load and integrate the Catalyst 1700 MIB extensions into Novell's NetWare Management System (NMS). These files are located on the Catalyst 1700 DOS formatted diskette that is included with the Catalyst 1700 hub. The diskette contains the *CAT1700.mib* and *rfc1317.mib* files, as well as NMS Profiles. These Profiles can be integrated into Novell NMS following the instructions below.

These instructions are specific to Novell NMS version 2.0. For other versions or for additional information, consult your Novell NMS documentation.

1. Copy the MIB files.

   Using the DOS copy command, transfer the *CAT1700.mib* and *rfc1317.mib* files to the current NMS directory. For a standard NMS installation, this directory is usually \nms\snmpmibs\current.

   The file *CAT1700.mib* contains the Catalyst 1700 MIB extensions written in ASN.1 language. The file rfc1317.mib, also written in ASN.1, contains definitions of managed objects for RS-232-like devices as defined in RFC-1317.

2. From the NMS main window, select the **Tools > SNMP MIB Compiler** command.

   Confirm this action by selecting **OK** to compile all current MIB definitions.

   For the compilation to succeed, any existing Browser windows must first be closed.

3. Select the **Tools > SNMP MIB Browser** command from the NMS main window to view and set the Catalyst 1700 MIB objects.

4. Select the **Add** button to invoke the Profile Editor for creating a new profile.

The names of all the available Catalyst 1700 scalar and table objects should now be listed in the Group Attribute Choices box of the Profile Editor window.

5 Choose the objects or tables to be added to the new Profile's Attribute Selection box as appropriate.

6 Enter a name and a description for the Profile, a community string and a display attribute as needed.

7 Click **Save** to store the named Profile to disk.

With a properly configured Catalyst 1700 on an accessible local network, you are now ready to browse and set the Catalyst 1700 objects that are included in this Profile.

8 Catalyst 1700 Enterprise Specific Traps Disposition.

To activate or de-activate the ability to receive and process traps at the NMS station, the **Fault>Alarm Disposition** command must be used. Refer to the NMS User's Guide for detailed information.

Using the Supplied NMS Profiles

The Catalyst 1700 diskette contains a number of NMS Profiles that have been set up with Catalyst 1700 MIB objects. These files are provided for convenience and can be used without change. Steps 1 through 8 listed above must be completed prior to the steps described below.

1 Copy the files **.PRF* to the directory *nms\snmp\mibs\profiles* for a standard NMS installation.

The supplied profiles are:

fssysinf.prf which contains the objects defined for the sysInfo group of the Catalyst 1700-specific MIB.

fssyscfg.prf which contains the sysConfig group.

fsport.prf which contains the swPortTable of the Port group.

fsrxstat.prf which is the swPortRxStatTable of the Port group.

fstxstat.prf which is the swPortTxStatTable of the Port group.

fsledmet.prf, which is the Profile that will plot a line graph showing the number of utilization LEDs currently lit (the value of the sysInfoUtilDisplay object).

fsupgrade.prf, which contains the upgrade group.

fsnetmgmt.prf, which contains the netMgmt scalar objects.

fssetcl.prf, which contains the netMgmtSetClientTable of the netMgmt group.

fstrapcl.prf, which contains the netMgmtTrapClientTable of the netMgmt group.

MIB Compilation in a SunNet Manager Environment

This section describes how to load, integrate and use the Catalyst 1700 MIB extensions with SunNet Manager. These files are archived in TAR format on the Catalyst 1700 UNIX diskette that is included with the Catalyst 1700 hub.

The diskette contains the *CAT1700.mib* and *rfc1317.mib* files, as well as their pre-compiled versions known as Schema files. These Schema files can be immediately integrated into SunNet Manager without requiring additional compilation, following the instructions below.

These instructions are specific to SunNet Manager version 2.1. For other versions or for additional information, consult your SunNet Manager documentation.

1. Insert the Catalyst 1700 UNIX diskette into the floppy drive of the SunNet Manager workstation.
2. Use TAR to extract the MIB files.

   SunNet Manager is typically installed in the default directory */opt/SUNWconn/snm* or in the directory designated by the environment variable $SNMHOME.

   Extract all files on the Catalyst 1700 diskette into the directory *$SNMHOME/agents* on the SunNet Manager workstation as follows:

   ```
   cd $SNMHOME/agents
   tar xvf /dev/fd0
   ```

   On some UNIX platforms, the name of the floppy device may be different. On SunOS 5.3, it may be necessary to stop the Volume Manager before the TAR diskette can be read. As the root user, type the following:

```
/etc/init.d/volmgt stop
cd $SNMHOME/agents
tar xvf /dev/rdiskette
/etc/init.dvolmgt start
```

These files include:

CAT1700.mib, which contains the Catalyst 1700 MIB extensions written in ASN.1 language.

rfc1317.mib, also written in ASN.1, which contains definitions of managed objects for RS-232-like devices as defined in RFC-1317.

In addition, it includes the following schemas, that is pre-compiled versions of the two MIBs in a format recognizable by SunNet Manager.

```
CAT1700.mib.schema
CAT1700.mib.oid
CAT1700.mib.traps
rfc1317.mib.schema
rfc1317.mib.oid
```

3 Set up the SunNet Manager SNMP target configuration file.

Locate the keyword *na.snmp.hostfile* in the file */etc/snm.conf*. This keyword points to the file the SNMP proxy agent and SNMP trap proxy use to obtain target-specific information. In this file, add an entry for each Catalyst 1700 that is to be managed. The relevant Schema file names to be specified are *CAT1700.mib.schema*, *CAT1700.mib.traps*, and *rfc1317.mib.schema*. Refer to the "SunNet Manager 2.1 Reference" manual for a detailed specification of the file named by the keyword *na.snmp.hostfile*.

4 Start the SunNet Manager console.

The SunNet Manager will now understand the Catalyst 1700 Enterprise-Specific MIB and trap Schemas and the RFC-1317 MIB Schema.

5 Manage the Catalyst 1700 via the SunNet Manager Discover Tool.

Use the SunNet Manager Discover tool to locate and map the Catalyst 1700. Once the Catalyst 1700 is discovered and added as an icon to the network map, set up the icon properties to process the Catalyst 1700 Schema and RFC-1317 Schema. These Schemas will be displayed with the names CATALYST1700-MIB and RFC1317-MIB, respectively, on the icon properties window.

6. Click on the boxes to the left of CATALYST1700-MIB and RFC1317-MIB to enable management using these Schemas.
7. Type **localhost** on the proxy line to use the local SNMP proxy agent.

   The Catalyst 1700 is now manageable using the various facilities provided by SunNet Manager including the Quick Dump, Data Report, Event Report and Set Request facilities.