CHAPTER 7

LightStream 2020 MIB Reference

Network management systems that use SNMP work with information stored in a Management Information Base (MIB). This chapter:

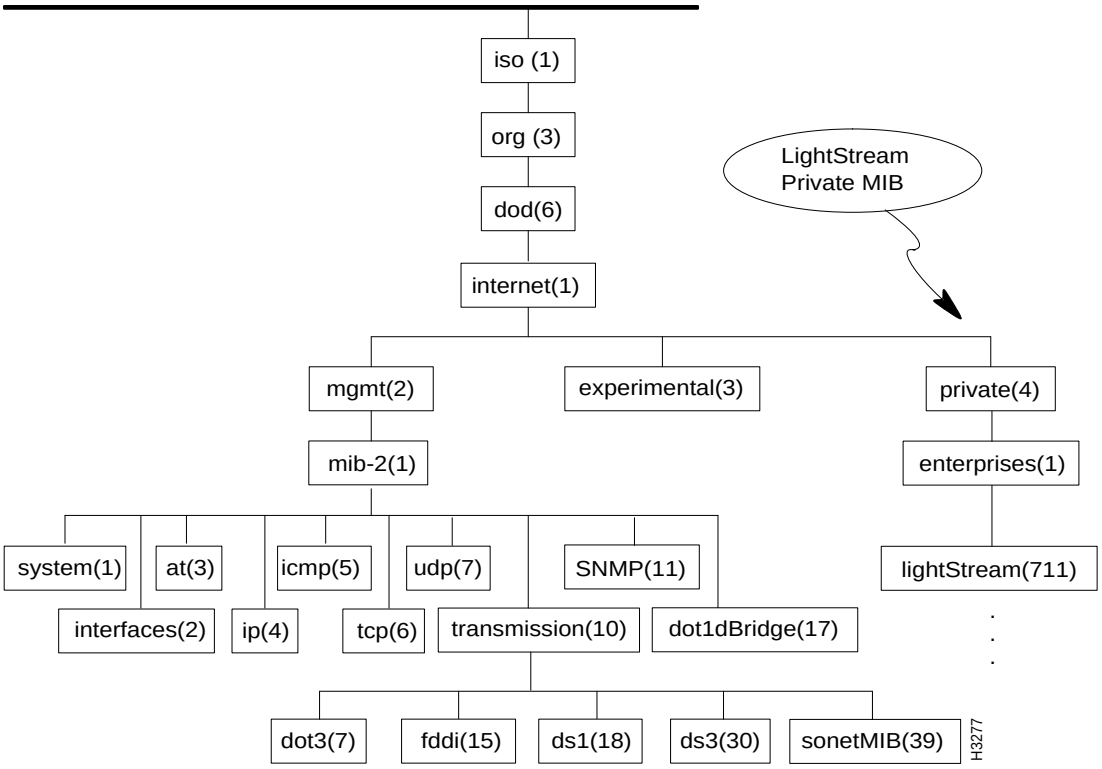
- Describes the MIB used to manage LightStream® 2020 enterprise ATM switches.
- Tells how to use this information to identify MIB objects by their addresses in the MIB. The SNMP commands getsnmp, getnextsnmp, setsnmp, and walksnmp require MIB addresses as arguments. These commands are described in Chapter 5.

Note MIB objects whose values may be read but not written are marked read only in the tables of this chapter. MIB objects that are not marked read only in the rightmost column of a table may be both read and written. (The table and table entry objects, like cardTable and cardEntry, are not accessible. They are shown here only because they help to structure the MIB.)

MIB Overview

The following figure shows a high-level diagram of the MIB that is used to manage LightStream switches. The organization of the MIB is defined in RFC 1155, a document called the *Structure of Management Information* (SMI).

Figure 7-1 MIB Structure

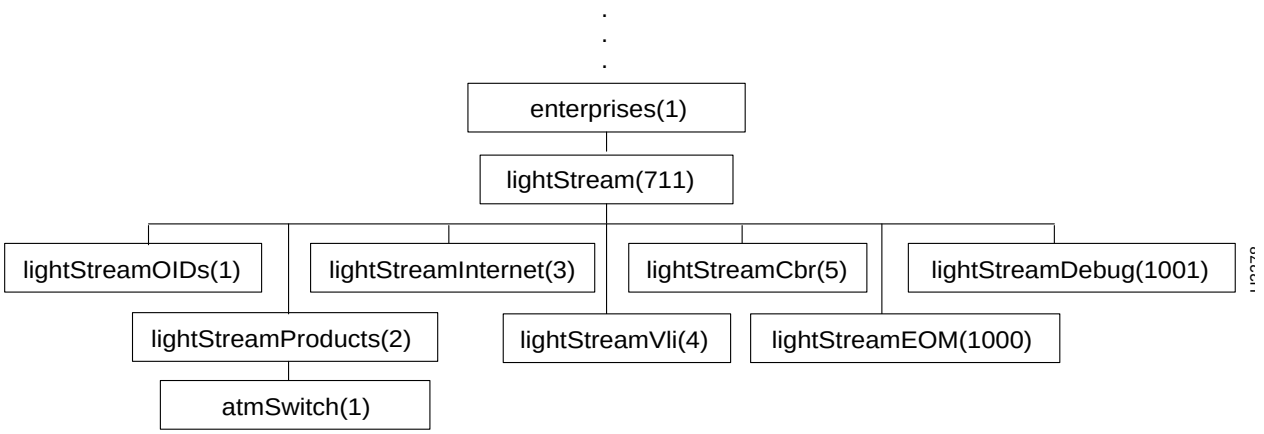


For information on some of the standard MIBs shown in the above figure, refer to the following documents:

- MIB-II (RFC 1213)
- Dot-3 Ethernet (RFC 1398)
- FDDI (RFC 1512)
- DS1 MIB (RFC 1232)
- DS3 MIB (RFC 1233)
- Dot-1 Bridge (RFC 1493)
- SONET (RFC 1595)

Under the Private subtree is the LightStream 2020 private MIB that is described in the remainder of this chapter. The high level structure of the LightStream 2020 private MIB is as follows:

Figure 7-2 The LightStream 2020 Private MIB



Note The ellipsis in this figure indicates where the upper levels of the MIB (the sequence of MIB objects described by the path iso.org.dod.internet.private) are not shown; see the previous figure for that information.

For detailed descriptions of those MIB objects that can be displayed and set as management objects, see the figures and tables in the following sections:

- The atmSwitch Subtree
- The lightStreamInternet Subtree
- The lightStreamVLI Subtree
- The lightStreamCbr Subtree

The lightStreamOIDs branch appears in the figure above and is displayed by the **browse** command, but is not described here because with one exception the objects in it are inaccessible (they cannot be displayed or set). The subbranches of the lightStreamOIDs branch are as follows:

lightStreamATM	Points to LightStream_ATM, defined in mib.c.
lsOther	Defines variable bindings in trap messages.
lsConfig	Identifies configuration-internal objects.

The two objects in the lsOther branch are lsTrapText and lsTrapName. These are display strings that are not accessible.

For information at a level of detail greater than can be given in this chapter, examine the MIB files themselves.

MIB Addresses

A MIB address identifies a particular MIB object by its location in the MIB. The four SNMP commands **getsnmp**, **getnextsnmp**, **setsnmp**, and **walksnmp** require MIB addresses in their arguments: These commands are described in Chapter .

Alternative Forms of MIB Addresses

You can specify the MIB address in a number of ways. For example, the following addresses all refer to the same MIB variable:

sysContact	A variable name.
1.3.6.1.2.1.1.4	A "path" of dot-separated numbers (the numbers in parentheses in the figures depicting the organization of the MIB).
mgmt.1.1.4	A combination, where the first element is a variable name and the numbers specify one branch in the subtree under it. Other examples: mib.1.4 and system.4.

Determining MIB Variable Names

To learn the names of the MIB variables in one of the subtrees shown in the above figures, you can specify the name of the subtree as the argument of the **walksnmp** command. This command displays all the "leaves" below the specified point in the MIB. You may also use the **browse** command to step through the branches of the MIB tree and, when you reach the end of a branch, to display the values of leaves one at a time.

Object Identifiers

The **getsnmp**, **getnextsnmp**, and **setsnmp** commands display and change the value of a single "leaf," that is, a single instance of a given MIB object. To identify a single instance of a MIB object at the end of a branch, you specify the object name followed by a numeric suffix known as an object identifier.

If there is only a single instance of the given object, then the identifier suffix for that instance is 0. For example, there is only one instance of the sysContact object, so its identifier is sysContact.0. The following command displays the name of the contact person for the target node:

```
cli> getsnmp sysContact.0
```

Just as with the SNMP commands and the **browse** command, described above, the first part of the object address, the MIB variable, may be specified in a number of ways. Thus, the following addresses all refer to the same MIB object:

- sysContact.0
- system.4.0
- mib.1.4.0
- mgmt.1.1.4.0
- 1.3.6.1.2.1.1.4.0

A single digit (0 in the example) is the simplest form of object identifier. For information about more complex types, see Port Identifiers, and see Multiply-Indexed Objects.

Note In the lists of MIB objects provided in this chapter, the object identifiers for table entries are indicated by the notation index=*object(s)* next to the table entry object. For example, under the CardTable object, next to the CardEntry object is the entry index=*card#*. Usually, the table entries are indexed by the first object(s) in the table itself. In the example just cited, entries in CardTable are indexed by CardIndex, the first table entry in the table.

Port Identifiers

The usual way of specifying a port number in the argument of a CLI command is in the format *c.p*, where *c* is the card (slot) number and *p* is the number of the port on that card. However, when a MIB object identified by port number is given as the argument of the **getsnmp**, **getnextsnmp**, or **setsnmp** command, a more complex format is used.

In effect, the long form of port number is *ctpp* or *cctpp*, where *c* is the card number, *p* is the port number, and *t* is the card type (2=FDDI, 1=Ether, 0=Other). The formula for determining the port number in this long format is as follows:

```
((c * 1000) + p + t)
```

The variables in this formula are as follows:

c — The card number, an integer in the range 1-11. A number in the range 1-10 is the number of the physical slot in which the card resides. The number 11 represents the logical interface for the LightStream ATM network.

p — The port number on the card, an integer in the range 0-7. If *c* is 11 (the logical interface for the LightStream ATM network) then *p* must be 4.

An offset indicating the card type. This is an arbitrary numeric value assigned to each type of card. In Release 2.0, the values of *k* are as follows:

100	Ethernet
200	FDDI
0	Other

For example, the MIB object that stores the alias or name for port 3 on card 4 is portInfoName.3004, or portInfoName.3104 if it is an Ethernet port, or portInfoName.3204 if it is an FDDI port.

Note Port number 1255 or 2255 refers to the control port (port 255) of the NP in slot 1 or slot 2, respectively.

As a second example, the following command tells us which data link connection management scheme is active on frame relay port 4.2 (card 4, port 2), specified here as port 4002:

```
cli> getsnmp frProvMiState.4002
Name: frProvMiState.4002
Value: 3
cli>
```

By looking this object up on page 18, or by searching for it in the MIB file private_mib.asn, we find that the value 3 means that the default ANSI standard T1.617 Annex D is active on this port.

Multiply-Indexed Objects

Some objects require us to specify more than one index number. For example, the following command tells us that the local LMI state of the frame relay circuit on DLCI 16, port 4, card 2 is inactive:

```
cli> getsnmp frCktInfoLclLMI.4002.16
Name: frCktInfoLclLMI.4002.16
Value: 2
cli
```

The double index 4002.16 means that this frCktInfoLclLMI object is indexed first by port number 4002 (equivalent to port 4.2) and then by DLCI number 16. For the fact that the value 2 means "inactive", see page 20, or see the description text for frCktInfoLclLMI in the private_mib.asn MIB file.

The following table shows examples of doubly-indexed entries in the LightStream private MIB:

Table Entry Object	Indexing Objects
frCktEntry	frCktSrcIfIndex, frCktSrcDlci
frCktInfoEntry	frCktInfoIfIndex, frCktInfoDlci
sUniCktEntry	sUniCktSrcIfIndex, sUniCktSrcVCI
sUniCktInfoEntry	sUniCktInfoIfIndex, sUniCktInfoVCI
pvcEntry	pvcSrcIfIndex, pvcSrcPvcId
lsFrameRelayDlciStatEntry	frameRelayDlciStatPortIndex, frameRelayDlciStatDlciIndex
lsEdgePortToSwMsgLenEntry	edgeToSwMsgLenPortIndex, edgeToSwMsgLenBinIndex
lsEdgeSwToPortMsgLenEntry	edgeToPortMsgLenPortIndex, edgeToPortMsgLenBinIndex
lsEdgeCpuWorkloadEntry	lsEdgeWorkloadCardIndex, lsEdgeWorkloadTypeIndex
lsFrameForwardStatEntry	frameForwardStatPortIndex, frameForwardStatIndex
lsTrunkCpuWorkloadEntry	lsTrunkWorkloadCardIndex, lsTrunkWorkloadTypeIndex
fsuPortXmtCellsEntry	fsuXmtCellsPortIndex, fsuXmtCellsPriorityIndex
fsuQueueCellLenEntry	fsuQueueCellLenPortIndex, fsuQueueCellLenSubQIndex
fsuDropEventEntry	fsuDropEventPortIndex, fsuDropEventWatermarkIndex
lsCellVciStatEntry	cellVciStatPortIndex, cellVciStatVciIndex
gidLineCardEntry	gidLineCardChassis, gidLineCardSlot
gidPortEntry	gidPortChassis, gidPortID
lightStreamBridgeFilter-Entry	lightStreamBrFilterId, lightStreamBrFilterTokenIndex

Table Entry Object	Indexing Objects
lightStreamBridgeFilter-ParameterEntry	lightStreamBrFilterParmPort, lightStreamBrFilterParmFilterId
lightStreamVliPortWork-GroupEntry	lightStreamVliPortWorkGroupPort, lightStreamVliPortWorkGroupID

Objects in standard MIBs may also be multiply indexed. These appear with more than one number in the object identifier when you walk or browse the MIB. You can also identify these objects by examining the MIB file in which they are defined: the INDEX entry of the definition lists more than one object name as indexing objects.

An example is the fddimibPATHEntry object defined in the fddi.asn MIB file. The FDDI stations implemented in the LightStream node are single-MAC, dual-attachment FDDI stations. There are two paths per SMT. (An FDDI path is the sequence of MACs and ports in token order.) The paths are managed by the fddimibPATHTable object. The fddimibPATHTable object is defined as a sequence of fddimibPATHEntry objects.

The fddimibPATHEntry object is doubly indexed. The first indexing object is fddimibPATHSMTIndex, and the second is fddimibPATHIndex. LightStream software calculates the values of these SMT index objects as follows:

- There are two SMTs per card, indexed as 2*c* and as (2*c*)-1, where *c* is the card number. For example, the card 3 SMT indices are 5 and 6, and the card 4 SMT indices are 7 and 8.
- The MAC for each SMT has 1 as its index. For example, the MAC on the first SMT on card 3 has 5.1 as its index.
- The two token-ring paths on each SMT are indexed 1 and 2. For example, the two paths on the first SMT on card 3 have 5.1 and 5.2 as their indices.
- The two physical ports for the logical interface on each SMT are also indexed 1 and 2. For example, the two ports for the first SMT on card 3 have 5.1 and 5.2 as their indices.

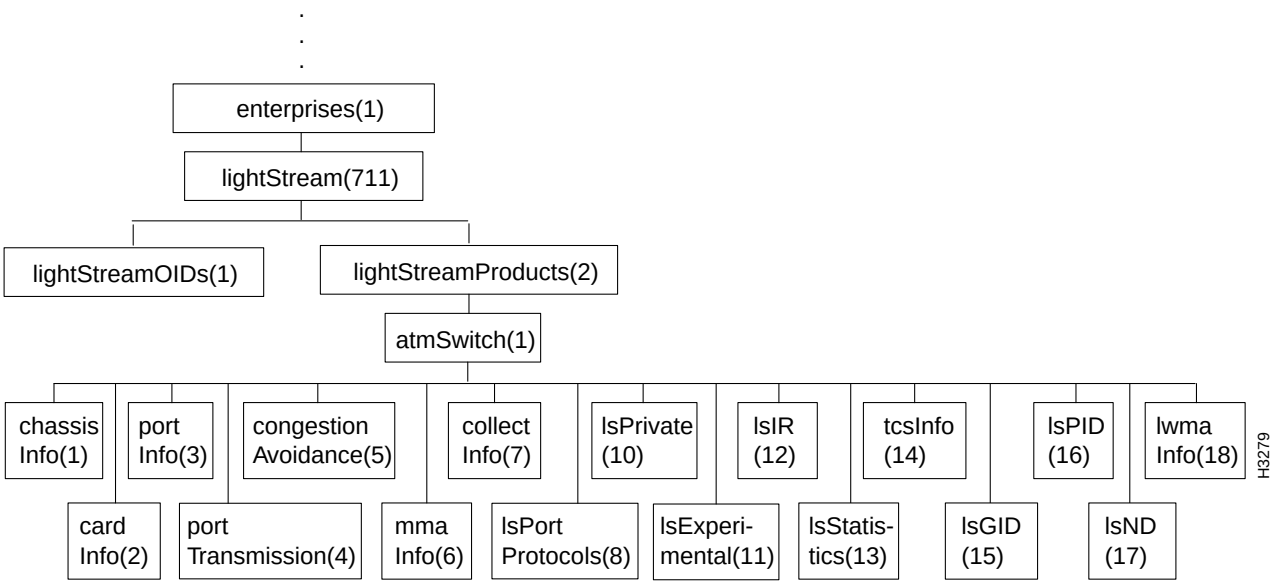
The following example shows the fddimibPATHTable objects for a LightStream node. These objects are defined for card 3 (SMT 5 and 6).

```
cli> walksnmp fddimibPATHTable
Name: fddimibPATHSMTIndex.5.1  Value: 5
Name: fddimibPATHSMTIndex.5.2  Value: 5
Name: fddimibPATHSMTIndex.6.1  Value: 6
Name: fddimibPATHSMTIndex.6.2  Value: 6
Name: fddimibPATHIndex.5.1     Value: 1
Name: fddimibPATHIndex.5.2     Value: 2
Name: fddimibPATHIndex.6.1     Value: 1
Name: fddimibPATHIndex.6.2     Value: 2
Name: fddimibPATHTVXLowerBound.5.1 Value: 2500000
Name: fddimibPATHTVXLowerBound.5.2 Value: 2500000
Name: fddimibPATHTVXLowerBound.6.1 Value: 2500000
Name: fddimibPATHTVXLowerBound.6.2 Value: 2500000
Name: fddimibPATHTMaxLowerBound.5.1 Value: 165000000
Name: fddimibPATHTMaxLowerBound.5.2 Value: 165000000
Name: fddimibPATHTMaxLowerBound.6.1 Value: 165000000
Name: fddimibPATHTMaxLowerBound.6.2 Value: 165000000
Name: fddimibPATHMaxTReq.5.1   Value: 165000000
Name: fddimibPATHMaxTReq.5.2   Value: 165000000
Name: fddimibPATHMaxTReq.6.1   Value: 165000000
Name: fddimibPATHMaxTReq.6.2   Value: 165000000
cli>
```

The atmSwitch Subtree

The following figure shows the atmSwitch subtree, which comprises most of the LightStream private MIB:

Figure 7-3 The atmSwitch Subtree



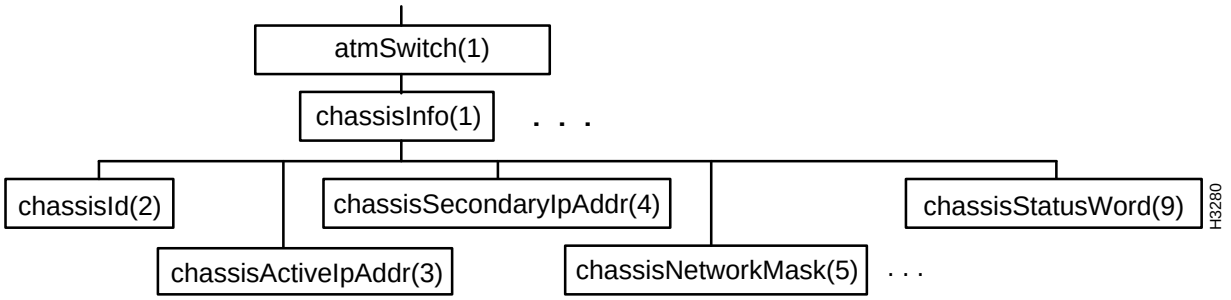
The branches of the atmSwitch subtree are as follows:

- The chassisInfo Branch
- The cardInfo Branch
- The portInfo Branch
- The portTransmission Branch
- The congestionAvoidance Branch
- The mmaInfo Branch
- The collectInfo Branch
- The lsPortProtocols Branch
- The lsPrivate Branch
- The lsExperimental Branch
- The lsIR Branch
- The lsStatistics Branch
- The tcsInfo Branch
- The lsGID Branch
- The lsPID Branch
- The lsND Branch
- The lwmaInfo Branch

The chassisInfo Branch

The figure below shows the chassisInfo branch of the atmSwitch subtree:

Figure 7-4 The chassisInfo Branch



Note The figure does not show all the objects in this branch because there are too many of them. Instead the first and last objects at each level are shown and an ellipsis (...) is used to indicate that additional objects exist. However, all the objects in this branch are listed in the following table.

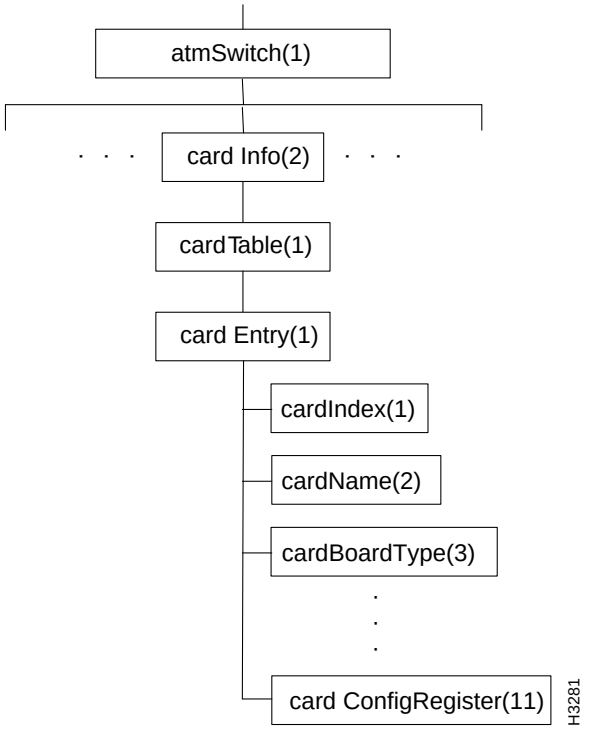
The following table shows the address and value of each object in the branch:

Objects in the chassisInfo Branch	MIB Branch	Value	
chassisId	chassisInfo.2	Integer (24 bits)	read only
chassisActiveIpAddr	chassisInfo.3	IpAddress	
chassisSecondaryIpAdd	chassisInfo.4	IpAddress	
chassisNetworkMask	chassisInfo.5	IpAddress	
chassisEthernetIpAddr	chassisInfo.6	IpAddress	
chassisEthernetIpMask	chassisInfo.7	IpAddress	
chassisDefaultIpRouter	chassisInfo.8	IpAddress	
chassisStatusWord	chassisInfo.9	Integer (bitmap)	read only
chassisConsoleTrapLevel	chassisInfo.10	1 = operational 2 = informational 3 = trace 4 = debug 5 = off	read only
chassisSwVersion	chassisInfo.11	Display string	read only
chassisTOD	chassisInfo.12	Display string	read only

The cardInfo Branch

The following figure shows the cardInfo branch of the atmSwitch subtree:

Figure 7-5 The cardInfo Branch



Note The figure does not show all the objects in this branch because there are too many of them. Instead the first and last objects at each level are shown and an ellipsis (...) is used to indicate that additional objects exist. However, all the objects in this branch are listed in the following table

The table that follows shows the address and value of each object in the branch. The objects in the cardInfo branch comprise a table, cardTable, in which each cardEntry row contains information about one card in the chassis.

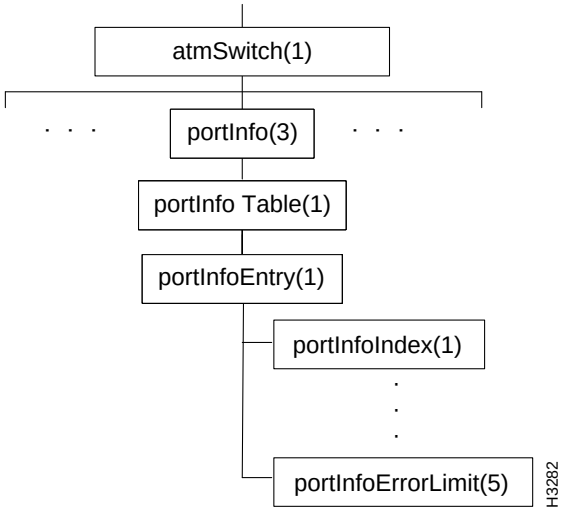
cardTable Objects	MIB Branch	Value	
cardTable	cardInfo.1		
cardEntry	cardTable.1	index=*card#*	
cardIndex	cardEntry.1	Integer (card #)	read only
cardName	cardEntry.2	DisplayString (19-char max)	
cardBoardType	cardEntry.3	DisplayString	read only
cardLcSoftwareVersion	cardEntry.4	DisplayString	read only
cardLccSoftwareVersion	cardEntry.5	DisplayString	read only
cardPID	cardEntry.6	Integer (PID)	read only

cardTable Objects	MIB Branch	Value	
cardMaxVCs	cardEntry.7	Integer (default 800)	
cardOperStatus	cardEntry.8	1 = up 2 = down 3 = testing 4 = empty	read only
cardAdminStatus	cardEntry.9	1 = up 2 = down 3 = testing	
cardStatusWord	cardEntry.10	Integer	read only
cardConfigRegister	cardEntry.11	1 = up 2 = down 3 = testing 4 = empty	read only
cardNumPorts	cardEntry.12	Integer	read only

The portInfo Branch

The following figure shows the portInfo branch of the atmSwitch subtree:

Figure 7-6 The portInfo Branch



The table that follows shows the address of each object in the branch and the type of each object. The objects in the portInfo branch comprise a table, portInfo-Table, in which each portInfoEntry row contains information about one port on a card.

PortInfoTable Objects	MIB Branch	Value	
portInfoTable	portInfo.1		
portInfoEntry	portInfoTable.1	index=*port#*	
portInfoIndex	portInfoEntry.1	Integer ((**c***1000)+*p*+*t*)	read only

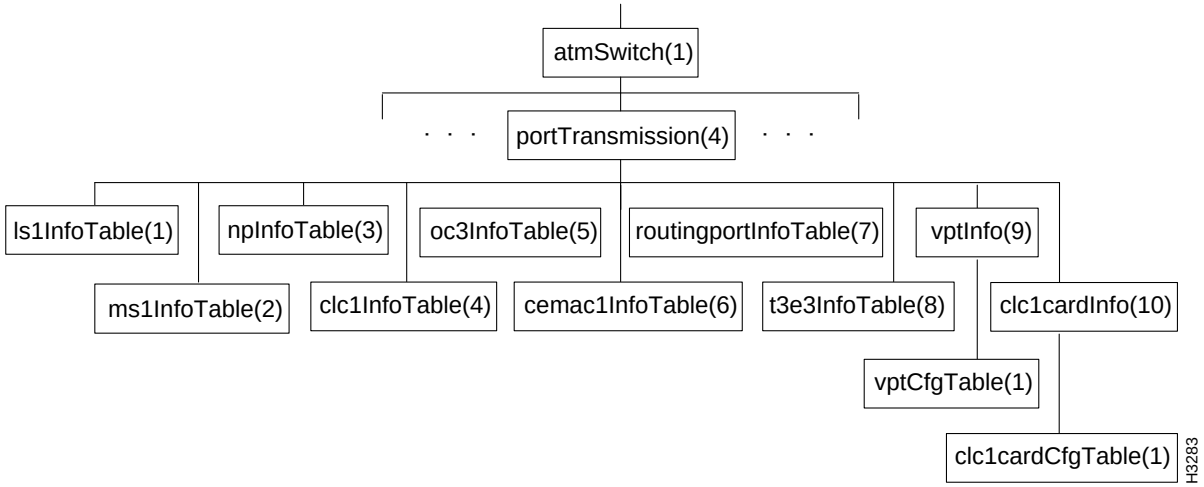
PortInfoTable Objects	MIB Branch	Value	
portInfoType	portInfoEntry.2	Integer	read only
		1 = empty	
		2 = error	
		3 = unknown	
		4 = switch	
		5= NP	
		6 = LS edge	
		7 = LS trunk	
		8 = MS trunk	
		10 =MS edge	
		11 = 2-port FDDI	
		12 = 3-port Ether	
		13 = token ring	
		14 = 8-port LS serial edge	
		15 = 8-port LS serial trunk	
		30 = generic CLC1	
		31 = 2-port OC3 edge	
		32 = 2-port OC3 trunk	
		33 = 8-port T3/E3 edge	
		34 = 8-port T3/E3 trunk	
		35 = 2-port 155 Mbit TAXI edge	
		36 = 2-port 155 Mbit TAXI trunk	
		37 = 8-port CBR edge	
portInfoSpecific	portInfoEntry.3	Object ID	read only
portInfoName	portInfoEntry.4	DisplayString (19-char max)	
PortInfoErrorLimit	portInfoEntry.5	Integer (10^{-n} rcv+xmt errors)	

lsEtherTable	portInfo.2		
lsEtherEntry	lsEtherTable.1	index=*port#*	
lsEtherMediaType	lsEtherEntry.1	Integer 1 = TPE 2 = AUI 3 = Fiber	read only

The portTransmission Branch

The following figure shows the portTransmission branch of the atmSwitch subtree:

Figure 7-7 The portTransmission Branch



Note The Figure does not show all the objects in this branch because there are too many of them. Instead the first and last objects at each level are shown and an ellipsis (...) is used to indicate that additional objects exist. However, all the objects in this branch are listed in the following tables.

The following tables show the address and type of each object in the branch. The objects in the portTransmission branch comprise three MIB object tables: ls1InfoTable for low speed line cards, ms1InfoTable for medium speed line cards, npInfoTable for NP cards, and clc1InfoTable for cell line cards. Each row of a table (ls1InfoEntry, ms1InfoEntry, or npInfoEntry) contains information about a port on a card.

ls1InfoTable Objects	MIB Branch	Value	
ls1InfoTable	portTransmission.1		
ls1InfoEntry	ls1InfoTable.1	index=*port#*	
ls1InfoIfIndex	ls1InfoEntry.1	Integer ((*c**1000)+*p*+*t*)	read only
ls1InfoType	ls1InfoEntry.2	Integer 1=V.35 2=RS-422 3=RS-530 4=T1 99=unknown	read only
ls1InfoOperCsuType	ls1InfoEntry.3	Integer 1=none 2=larse	read only
ls1InfoAdminCsuType	ls1InfoEntry.4	Integer 1=none 2=larse	

ls1InfoTable Objects	MIB Branch	Value	
ls1InfoOperRcvBaudRate	ls1InfoEntry.5	Integer	read only
ls1InfoAdminRcvBaudRate	ls1InfoEntry.6	Integer	
ls1InfoOperXmitBaudRate	ls1InfoEntry.7	Integer	read only
ls1InfoAdminXmitBaudRate	ls1InfoEntry.8	Integer	
ls1InfoOperNetIntType	ls1InfoEntry.9	Integer 1=DCE 2=DTE 3=DCE with local tt clock loop	read only
ls1InfoAdminNetIntType	ls1InfoEntry.10	Integer 1=DCE 2=DTE 3=DCE with local tt clock loop	
ls1InfoOperModemState	ls1InfoEntry.13	Integer(Bitmask) If port is DTE: 0=DCD, 1=DSR If port is DCE: 0=RTS, 1=DTR	read only
ls1InfoOperProtocol	ls1InfoEntry.15	Integer 1 = trunk 2 = FR 3 = FF 4 = PPP 5 = unkown	read only
ls1InfoAdminProtocol	ls1InfoEntry.16	Integer 1 = trunk 2 = FR 3 = FF 4 = PPP 5 = unknown	
ls1InfoOperControlBandwidthSize	ls1InfoEntry.21	Integer	read only
ls1InfoAdminControlBandwidthSize	ls1InfoEntry.22	Integer	
ls1InfoOperDataBandwidthSize	ls1InfoEntry.23	Integer	read only
ls1InfoAdminDataBandwidthSize	ls1InfoEntry.24	Integer	
ls1InfoOperLoopMode	ls1InfoEntry.25	Integer 1 = none 2 = internal 3 = external 4 = remote	read only
ls1InfoAdminLoopMode	ls1InfoEntry.26	Integer 1 = none 2 = internal 3 = external 4 = remote	
ls1InfoLcAutoEnable	ls1InfoEntry.27	Integer 1 = enable 2 = disable	read only
ls1InfoLcDebugLevel	ls1InfoEntry.28	Integer	read only

ls1InfoTable Objects	MIB Branch	Value	
ls1InfoDataCellCapacity	ls1InfoEntry.29	Integer	read only
ls1InfoDataCellAvailable	ls1InfoEntry.30	Integer	read only
ls1InfoMeasuredBaudRate	ls1InfoEntry.31	Integer	read only
ls1InfoLinkUtilization	ls1InfoEntry.32	Integer	read only
ls1InfoAdminOperTrigger	ls1InfoEntry.99	Integer 1=set admin values 99=other	

ms1InfoTable Objects	MIB Branch	Value	
ms1InfoTable	portTransmission.2		
ms1InfoEntry	ms1InfoTable.1	index=*port#*	
ms1InfoIfIndex	ms1InfoEntry.1	Integer ((*c**1000)+*p*+*t)*	read only
ms1InfoOperCableLength	ms1InfoEntry.2	Integer 1=0-450 ft T3 2=450-900 ft T3 3=0-400 ft E3 4=300-1000 ft E3 5=800-1300 ft E3 6=1100-1900 ft E3	read only
ms1InfoAdminCableLength	ms1InfoEntry.3	Integer 1=0-450 ft T3 2=450-900 ft T3 3=0-400 ft E3 4=300-1000 ft E3 5=800-1300 ft E3 6=1100-1900 ft E3	
ms1InfoOperProtocol	ms1InfoEntry.4	Integer 1=trunk 2=ATM UNI 3=unknown	read only
ms1InfoAdminProtocol	ms1InfoEntry.5	Integer 1=trunk 2=ATM UNI 3=unknown	
ms1InfoOperControlBandwidthSize	ms1InfoEntry.10	Integer	read only
ms1InfoAdminControlBandwidthSize	ms1InfoEntry.11	Integer	
ms1InfoOperDataBandwidthSize	ms1InfoEntry.12	Integer	read only
ms1InfoAdminDataBandwidthSize	ms1InfoEntry.13	Integer	
msInfoLcAutoEnable	ms1InfoEntry.14	Integer 1=enable 2=disable	read only
msInfoLcDebugLevel	ms1InfoEntry.15	Integer	read only
ms1InfoOperScramble	ms1InfoEntry.16	Integer 1=enable 2=disable	read only

ms1InfoTable Objects	MIB Branch	Value	
ms1InfoAdminScramble	ms1InfoEntry.17	Integer 1=enable 2=disable	
ms1InfoDataCellCapacity	ms1InfoEntry.18	Integer	read only
ms1InfoDataCellAvailable	ms1InfoEntry.19	Integer	read only
ms1InfoLinkUtilization	ms1InfoEntry.20	Integer	read only
ms1InfoOperFraming	ms1InfoEntry.21	Integer 1=Cell Payload Scrambling 2=T3 HEC 3=G.804	read only
ms1InfoAdminOperTrigger	ms1InfoEntry.99	Integer 1=set admin values 99=other	

npInfoTable Objects	MIB Branch	Value	
npInfoTable	portTransmission.3		
npInfoEntry	npInfoTable.1	index=*port#*	
npInfoIfIndex	npInfoEntry.1	Integer ((*c**1000)+*p*+*t*)	read only
npInfoIPCommittedRate	npInfoEntry.5	Integer	
npInfoIPCommittedBurst	npInfoEntry.6	Integer	
npInfoIPExcessRate	npInfoEntry.7	Integer	
npInfoIPExcessBurst	npInfoEntry.8	Integer	
npInfoIPNCircuits	npInfoEntry.9	Integer	
npInfoAdminOperTrigger	npInfoEntry.99	Integer 1=set admin values 99=other	

clc1InfoTable Objects	MIB Branch	Value	
clc1InfoTable	portTransmission.4		
clc1InfoEntry	clc1InfoTable.1	index=*port#*	
clc1InfoIfIndex	clc1InfoEntry.1	Integer ((*c**1000)+*p*+*t*)	read only
clc1InfoOperProtocol	clc1InfoEntry.4	Integer 1=trunk 2=ATM UNI 3=unknown	read only
clc1InfoAdminProtocol	clc1InfoEntry.5	Integer 1=trunk 2=ATM UNI 3=unknown	

clc1InfoTable Objects	MIB Branch	Value	
clc1InfoOperLoopMode	clc1InfoEntry.6	Integer 1=none 2=internal 3=external	read only
clc1InfoAdminLoopMode	clc1InfoEntry.7	Integer 1=none 2=internal 3=external	
clc1InfoOperControlBandwidthSize	clc1InfoEntry.10	Integer	read only
clc1InfoAdminControlBandwidthSize	clc1InfoEntry.11	Integer	
clc1InfoOperDataBandwidthSize	clc1InfoEntry.12	Integer	read only
clc1InfoAdminDataBandwidthSize	clc1InfoEntry.13	Integer	
clcInfoLcAutoEnable	clc1InfoEntry.14	Integer 1=enable 2=disable	read only
clcInfoLcDebugLevel	clc1InfoEntry.15	Integer	read only
clc1InfoOperScramble	clc1InfoEntry.16	Integer 1=enable 2=disable	read only
clc1InfoAdminScramble	clc1InfoEntry.17	Integer 1=enable 2=disable	
clc1InfoDataCellCapacity	clc1InfoEntry.18	Integer	read only
clc1InfoDataCellAvailable	clc1InfoEntry.19	Integer	read only
clc1InfoLinkUtilization	clc1InfoEntry.20	Integer	read only
clc1InfoOperClock	clc1InfoEntry.21	Integer 1=internal 2=external	read only
clc1InfoAdminClock	clc1InfoEntry.22	Integer 1=internal 2=external	
clc1InfoAdminOperTrigger	clc1InfoEntry.99	Integer 1=set admin values 99=other	

oc3InfoTable Objects	MIB Branch	Value	
oc3InfoTable	portTransmission.5		
oc3InfoEntry	oc3Infotable.1	index=*port#*	
oc3InfoReceiveSignalDetect	oc3InfoEntry.1	Integer 1=active 2=inactive	read only
oc3InfoTransmitSafetySwitch	oc3InfoEntry.2	Integer 1=enable 2=disable 3=multimode	read only

oc3InfoTable Objects	MIB Branch	Value
oc3InfoMediumType	oc3InfoEntry.3	Integer 1=SONET 2=SDH

Note: sonetMediumType in the standard MIB is a read-only object, but oc3InfoMediumType here is writeable.

npInfoTable Objects	MIB Branch	Value	
cemacInfoTable		portTransmission.6	
cemacInfoEntry	cemacInfoTable.1	index=*port#*	
cemacInfoDataBandwidth	cemacInfoEntry.2	Integer	read only
cemacInfoPortType	cemacInfoEntry.3	Integer 1=T1 circuit emul. (def.) 2=E1 circuit emul. 3=I15M 4=I63M 5=PBX 6=transcoder 7=I192 8=Y15M 9=Y63M 10=T1 11=X21 12=RS-449 13=V.35	read only
cemacInfoClkMode	cemacInfoEntry.4	Integer 1=Adaptive (default) 2=SRTS 3=synchronous	
cemacInfoLineBuildOut	cemacInfoEntry.5	Integer 1=-7.5 dB 2=-15.0 dB 3=0.0 dB 4=0-133 ft. (default) 5=133-266 ft. 6=266-399 ft. 7=399-533 ft. 8=533-655 ft.	
cemacInfoServiceType	cemacInfoEntry.6	Integer 1=circuit emulation (default) 2=fractional	read only

npInfoTable Objects	MIB Branch	Value
routingportInfoTable	portTransmission.7	
routingportInfoEntry	routingportInfoTable.1	index=*port#*
routingportInfoSupportsCdvVc	routingportInfoEntry.1	Integer 1=supported 2=not supported
routingportInfoMaxPortCdv	routingportInfoEntry.2	Integer (ms)

npInfoTable Objects	MIB Branch	Value	
routingportInfoExpectedPropDelay	routingportInfoEntry.3	Integer (ms)	
routingportInfoExpectedConstDelay	routingportInfoEntry.4	Integer (ms)	read only
routingportInfoCdvBwAvailable	routingportInfoEntry.5	Integer	read only
routingportInfoAdminWeight	routingportInfoEntry.6	Integer	
routingportInfoAdditiveCdv	routingportInfoEntry.7	Integer	read only

npInfoTable Objects	MIB Branch	Value
t3e3InfoTable	portTransmission.8	
t3e3InfoEntry	t3e3InfoTable.1	index=*port#*
t3e3InfoCableLen	t3e3InfoEntry.1	Integer 1=0-450 ft. T3 2=450-900 ft. T3 3=0-400 ft E3 4=300-1000 ft. E3 5=800-1300 ft. E3 6=1100-1900 ft. E3
t3e3InfoFraming	t3e3InfoEntry.2	Integer (ms) 1=cell payload scrambling 2=T3 HEC 3=G.804

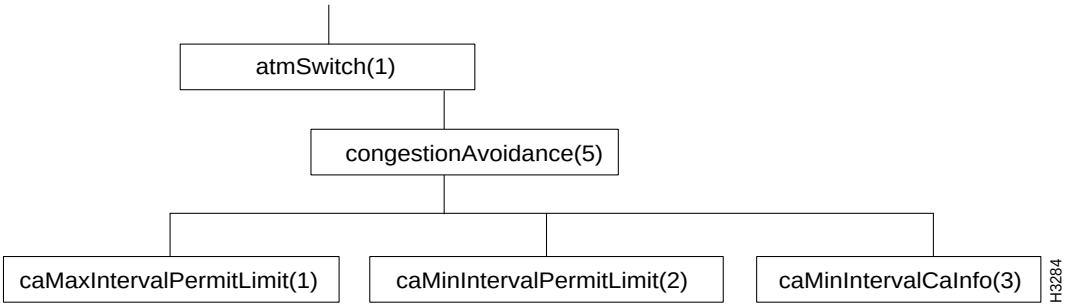
npInfoTable Objects	MIB Branch	Value	
vptInfo	portTransmission.9		
vptCfgTable	vptInfo.1		
vptEntry	vptCfgTable.1	index= *port#*.vptVirtualTrunkId	
vptVirtualTrunkId	vptEntry.1	Integer (0-255)	read only
vptVPI	vptEntry.2	Integer (0-255) 0=disable VPI selection	

npInfoTable Objects	MIB Branch	Value
clc1cardInfo	portTransmission.10	
clc1cardCfgTable	clc1cardInfo.1	
clc1cardEntry	clc1cardCfgTable.1	index=*card#*
clc1cardRate	clc1cardEntry.1	Integer (0-446000)

The congestionAvoidance Branch

The following figure shows the congestionAvoidance branch of the atmSwitch subtree:

Figure 7-8 The congestionAvoidance Branch



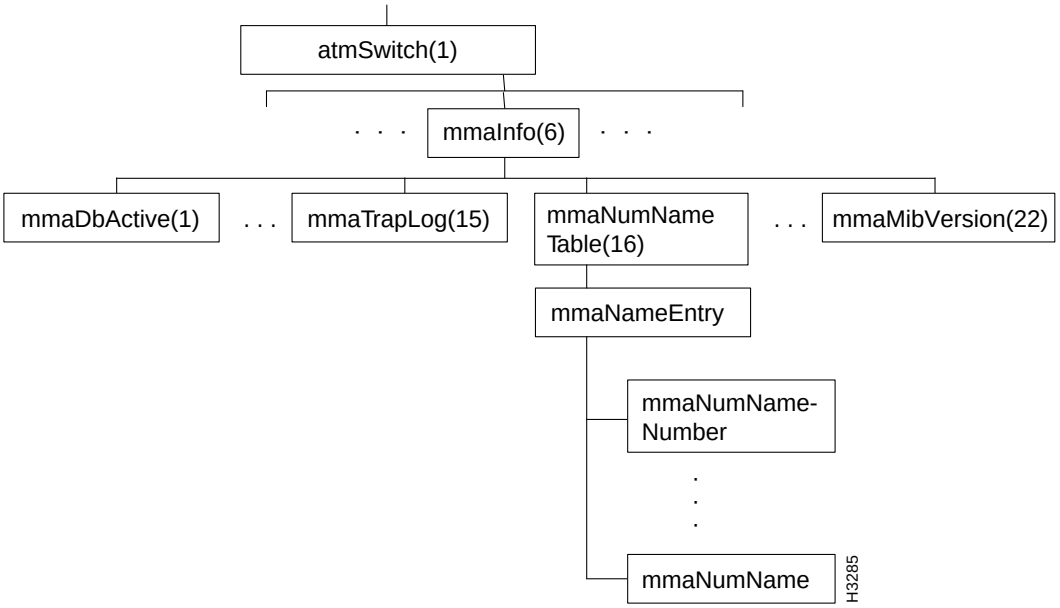
The table that follows shows the Address and type of each object in the branch.

congestionAvoidance Objects	MIB Branch	Value
caMaxIntervalPermitLimit	congestionAvoidance.1	Integer
caMinIntervalPermitLimit	congestionAvoidance.2	Integer
caMinIntervalCaInfo	congestionAvoidance.3	Integer

The mmaInfo Branch

The following figure shows the mmaInfo branch of the atmSwitch subtree:

Figure 7-9 The mmaInfo Branch



Note The figure does not show all the objects in this branch because there are too many of them. Instead the first and last objects at each level are shown and an ellipsis (...) is used to indicate that additional objects exist. However, all the objects in this branch are listed in the following tables.

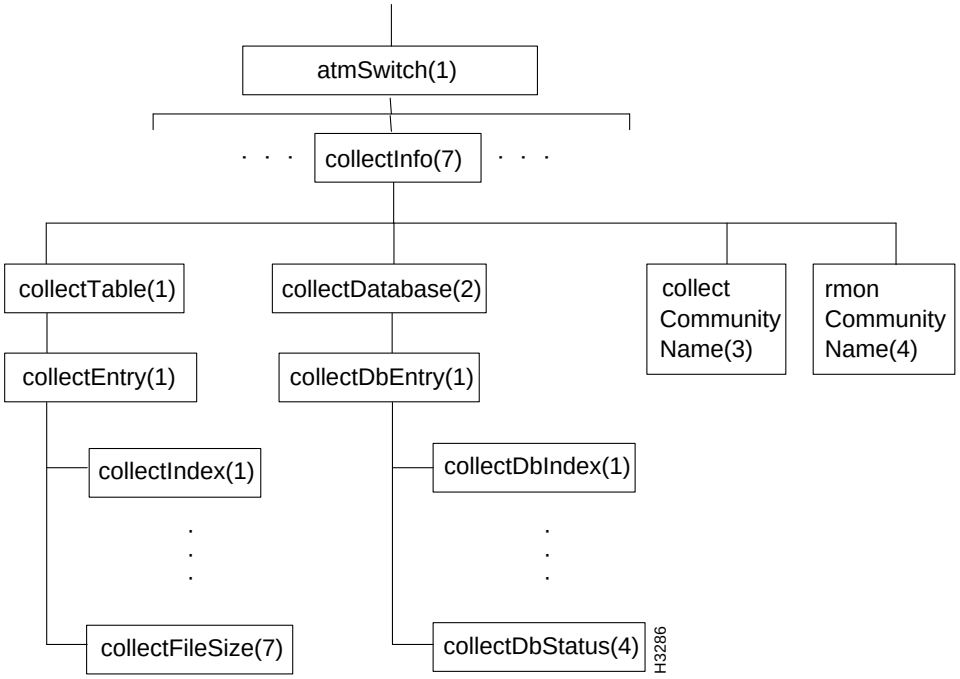
The following tables show the address and type of each object in the branch. The mmacardInfo branch includes three tables, mmaFtTable, mmaTrapCommunityTable, and mmaNumNameTable.

mmaInfo Objects	MIB Branch	Value	
mmaDbActive	mmaInfo.1	Integer (not used)	
mmaTrapFilter	mmaInfo.2	Integer 1=operational 2=informational 3=trace 4=debug	
mmaTrapLanguage	mmaInfo.3	Integer (1=English)	
mmaCollectionSpace	mmaInfo.4	Integer (Kbytes)	
mmaConfigHost	mmaInfo.5	Octet String	
mmaConfigAuthor	mmaInfo.6	Octet String	
mmaConfigID	mmaInfo.7	Integer	
mmaSetLock	mmaInfo.8	Integer 1=unlocked 2=locked (volatile) 3=locked (permanent)	
mmaPID	mmaInfo.9	Integer (PID)	read only
mmaTrapLog	mmaInfo.10	Integer 1=enable 2=disable	
mmaTrapNumber	mmaInfo.13	Integer	
mmaTrapOnOffState	mmaInfo.14	Integer 1=on 2=off 3=enable 4=disable	
mmaNumNameTable	mmaInfo.16		
mmaNumNameEntry	mmaNumNameTable.1	index=*chassisID*	
mmaNumNameNumber	mmaNumNameEntry.1	Integer (chassis ID)	read only
mmaNumName	mmaNumNameEntry.2	Display String (chassis name, 39-char max)	read only
mmaDumpMemStats	mmaInfo.17	Integer	
mmaDumpProfile	mmaInfo.18	Integer	
mmaDumpElog	mmaInfo.19	Integer	
mmaLwmpTimeouts	mmaInfo.20	Counter	
mmaLwmpPdusOutstandingMax	mmaInfo.21	Counter	
mmaMibVersion	mmaInfo.22	Integer (2=Version 2)	read only

The collectInfo Branch

The following figure shows the collectInfo branch of the atmSwitch subtree:

Figure 7-10 The collectInfo Branch



Note The figure does not show all the objects in this branch because there are too many of them. Instead the first and last objects at each level are shown and an ellipsis (...) is used to indicate that additional objects exist. However, all the objects in this branch are listed in the following table.

The following table shows the address and type of each object in the branch. The collectInfo branch contains two tables, collectTable and collectDatabase. Each row of each table contains information about about a collection that has been defined. The collectInfo branch also contains the collectCommunityName and rmonCommunityName objects (the latter not currently used).

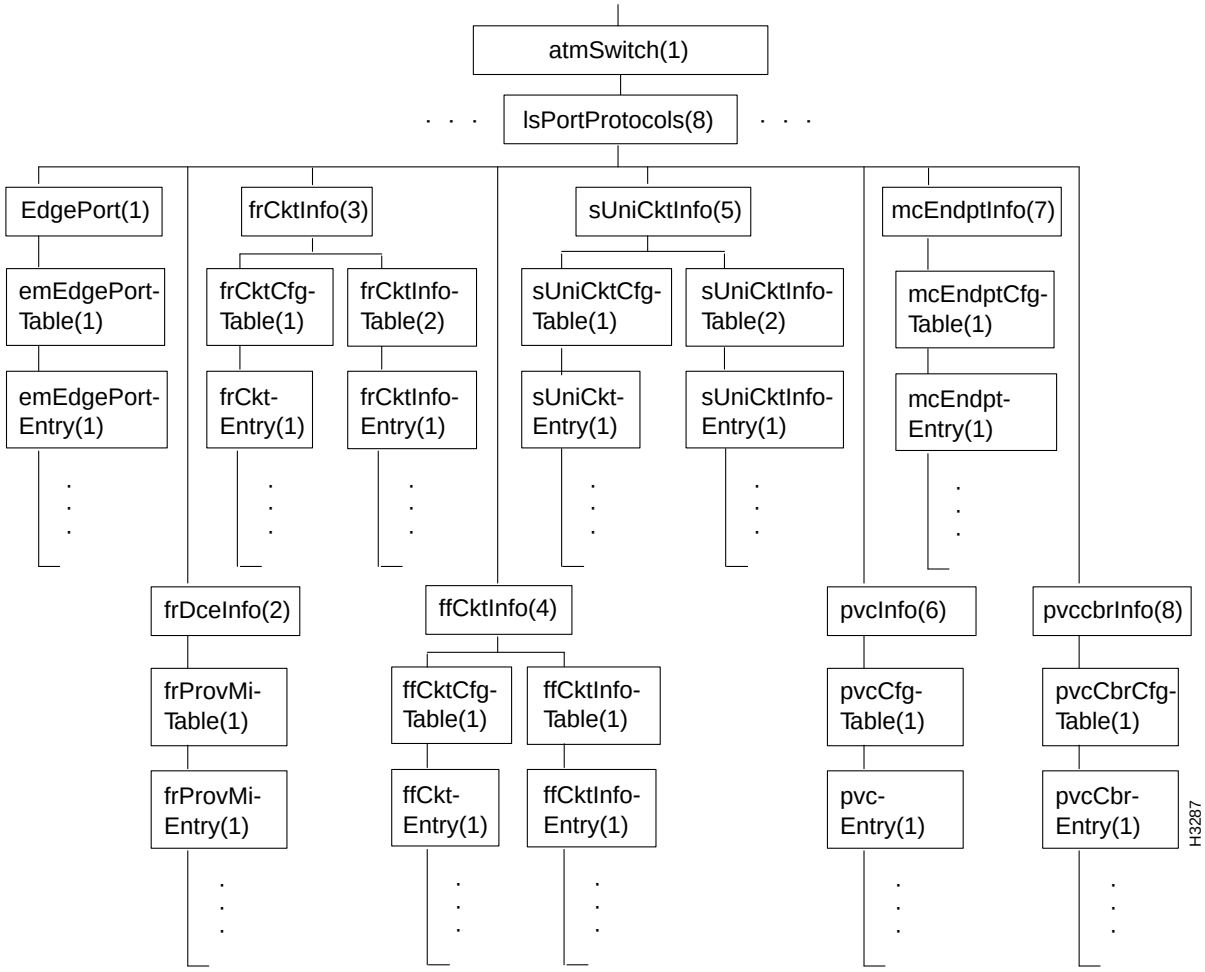
Objects in the collectInfo Branch	MIB Branch	Value	
collectTable	collectInfo.1		
collectEntry	collectTable.1	index=*collection#*	
collectIndex	collectEntry.1 collection number	Integer	read only
collectStatus	collectEntry.2	Integer 1=valid 2=create request 3=under creation 4=invalid	
collectStart	collectEntry.3	Integer (TOD sec)	

Objects in the collectInfo Branch	MIB Branch	Value	
collectFinish	collectEntry.4	Integer (TOD sec)	
collectInterval	collectEntry.5	Integer (sec)	
collectFileName	collectEntry.6	DisplayString	read only
collectFileSize	collectEntry.7	Integer (Kbytes)	
collectOperStatus	collectEntry.8	Integer 1=waiting 2=running 3=under creation	
collectDatabase	collectInfo.2		
collectDbEntry	collectDatabase.1	index=*collection#*.collectDBInstance	
collectDBIndex	collectDbEntry.1	Integer	read only
collectDBInstance	collectDbEntry.2	Integer	read only
collectDBObjectID	collectDbEntry.3	Object ID	
collectDBStatus	collectDbEntry.4	Integer 1=valid 2=create request 3=under creation 4=invalid	
collectCommunityName	collectInfo.3	DisplayString	
rmonCommunityName	collectInfo.4	DisplayString	

The lsPortProtocols Branch

The following figure shows the lsPortProtocols branch of the atmSwitch subtree:

Figure 7-11 The lsPortProtocols Branch



Note The figure does not show all the objects in this branch because there are too many of them. Instead the first and last objects at each level are shown and an ellipsis (...) is used to indicate that additional objects exist. However, all the objects in this branch are listed in the following tables.

The following tables show the address and type of each object in the branch. The lsPortProtocols branch contains several object tables containing information on the various services available on a LightStream network.

edgePortTable Objects	MIB Branch	Value
edgePort	lsPortProtocols.1	
edgePortTable	edgePort.1	
edgePortEntry	edgePortTable.1	index=*card#*

edgePortTable Objects	MIB Branch	Value	
edgeIfIndex	edgePortEntry.1	Integer ((*c**1000)+*p*+*t*)	read only
edgeUpcType	edgePortEntry.2	Integer default 1= ANSI	
edgeUserDataPerCell	edgePortEntry.3	Integer (bits) 1-384 default 341	
edgeCellDelayVariance	edgePortEntry.4	Integer (ms)	
edgePrincipalScale	edgePortEntry.5	Integer default 100 (100%)	
edgeSecondaryScale	edgePortEntry.6	Integer default 2 (2%)	
edgeMeteringFactor	edgePortEntry.7	Integer default 64 (4x)	
edgeMeteringBurstSize	edgePortEntry.8	Integer (cells) default 5	
edgeCallSetupRetry	edgePortEntry.9	Integer (sec) 1-3600 default 5	
edgeCallSetupBackoff	edgePortEntry.10	Integer (2^n) 1-10 default 5	
edgeMaxFrameSize	edgePortEntry.11	Integer (bytes) 48-8152 default 1516	

frProvMiTable Objects	MIB Branch	Value	
frDceInfo	lsPortProtocols.2		
frProvMiTable	frDceInfo.1		
frProvMiEntry	frProvMiTable.1	index=*card#*	
frProvMiIfIndex	frProvMiEntry.1	Integer ((*c**1000)+*p*+*t*)	read only
frProvMiState	frProvMiEntry.2	Integer 1=no LMI 2=FRIF 3=ANSI T1 (default) 4=CCITT Q.933	
frProvMiAddressLen	frProvMiEntry.3	Integer (octets) 2=two octets	
frProvMiNetRequestInterval	frProvMiEntry.4	Integer (sec) 5-30 (default 15)	
frProvMiNetErrorThreshold	frProvMiEntry.5	Integer 1-10 (default 3)	
frProvMiNetMonitoredEvents	frProvMiEntry.6	Integer 1-10 (default 4)	
frProvMiMaxSupportedVCs	frProvMiEntry.7	Integer	
frProvMiMulticast	frProvMiEntry.8	Integer 1=non-broadcast	
frProvMiUser-PollingInterval	frProvMiEntry.9	Integer (sec) 5-30 (default 10)	

frProvMiTable Objects	MIB Branch	Value
frProvMiUserFull-EnquiryInterval	frProvMiEntry.10	Integer 1-255 (default 6)
frProvMiUserError-Threshold	frProvMiEntry.11	Integer 1-10 (default 3)
frProvMiUser-MonitoredEvents	frProvMiEntry.12	Integer 1-10 (default 4)
frProvMiNet-InterfaceType	frProvMiEntry.13	Integer 1=UNI 2=NNI
frProvMiCompliance	frProvMiEntry.14	Integer 1=strict (default) 2=loose

Objects in the lsPortProtocols Branch	MIB Branch	Value	
frCktInfo	lsPortProtocols.3		
frCktCfgTable	frCktInfo.1		
frCktEntry	frCktCfgTable.1	index= *srcport#.srcDLCI*	
frCktSrcNode	frCktEntry.1	Integer (ID)	read only
frCktSrcIfIndex	frCktEntry.2	Integer ((*c**1000)+*p*+*t*)	read only
frCktSrcDlci	frCktEntry.3	LightStreamDLCI	read only
frCktAdminDestNode	frCktEntry.10	Integer (ID)	
frCktOperDestNode	frCktEntry.11	Integer (ID)	read only
frCktAdminDestIfIndex	frCktEntry.12	Integer ((*c**1000)+*p*+*t*)	read only
frCktOperDestIfIndex	frCktEntry.13	Integer ((*c**1000)+*p*+*t*)	read only
frCktAdminDestDlci	frCktEntry.14	LightStreamDLCI	
frCktOperDestDlci	frCktEntry.15	LightStreamDLCI	read only
frCktAdminSrcInsuredRate	frCktEntry.25	Integer (bits/s) default 0	
frCktOperSrcInsuredRate	frCktEntry.26	Integer (bits/s) default 0	read only
frCktAdminSrcInsuredBurst	frCktEntry.27	Integer (bytes) default 0	
frCktOperSrcInsuredBurst	frCktEntry.28	Integer (bytes) default 0	read only
frCktAdminSrcMaxRate	frCktEntry.29	Integer (bits/s) default: line rate	
frCktOperSrcMaxRate	frCktEntry.30	Integer (bits/s) default: line rate	read only
frCktAdminSrcMaxBurst	frCktEntry.31	Integer (bytes) default 0	
frCktOperSrcMaxBurst	frCktEntry.32	Integer (bytes) default 0	read only

Objects in the lsPortProtocols Branch	MIB Branch	Value	
frCktOperDestInsuredRate	frCktEntry.34	Integer (bits/s) default 0	read only
frCktOperDestInsuredBurst	frCktEntry.36	Integer (bytes) default 0	read only
frCktOperDestMaxRate	frCktEntry.38	Integer (bits/s) default: line rate	read only
frCktOperDestMaxBurst	frCktEntry.40	Integer (bytes) default 0	read only
frCktOperSecondaryScale	frCktEntry.41	Integer default 255	read only
frCktAdminSecondaryScale	frCktEntry.42	Integer default 255	
frCktOperPrinBwType	frCktEntry.43	Integer 1=guaranteed (default) 2=insured	read only
frCktAdminPrinBwType	frCktEntry.44	Integer 1=guaranteed (default) 2=insured	
frCktOperTransPri	frCktEntry.45	Integer 0 (default), 1	read only
frCktAdminTransPri	frCktEntry.46	Integer 0 (default), 1	
frCktOperUserDataPerCell	frCktEntry.47	Integer (bits) 0-384 (default 0)	read only
frCktAdminUserDataPerCell	frCktEntry.48	Integer (bits) 0-384 (default 0)	
frCktStatus	frCktEntry.99	Integer 1=enable 2=disable (default) 3=delete	

rCktInfoTable Objects	MIB Branch	Value	
frCktInfoTable	frCktInfo.2		
frCktInfoEntry	frCktInfoTable.1	index=*port#.DLCI#*	
frCktInfoIfIndex	frCktInfoEntry.1	Integer ((*c**1000)+*p*+*t*)	read only
frCktInfoDlci	frCktInfoEntry.2	LightStreamDLCI	read only
frCktInfoLclLMI	frCktInfoEntry.3	Integer 1=active 2=inactive	read only
frCktInfoRmtLMI	frCktInfoEntry.4	Integer 1=active 2=inactive	read only
frCktInfoCallIDIncoming	frCktInfoEntry.5	Integer (ID)	read only
frCktInfoCallIDOutgoing	frCktInfoEntry.6	Integer (ID)	read only

rCktInfoTable Objects	MIB Branch	Value	
frCktInfoDownstreamState	frCktInfoEntry.7	Integer 1=active 2=inactive	read only
frCktInfoUpstreamState	frCktInfoEntry.8	Integer 1=active 2=inactive	read only
frCktInfoLastAtmErr	frCktInfoEntry.9	Octet String	read only
frCktInfoDataCellsRequired	frCktInfoEntry.10	Integer	read only
frCktInfoLastAtmLocation	frCktInfoEntry.11	Display String	read only

ffCktCfgTable Objects	MIB Branch	Value	
ffCktInfo	lsPortProtocols.4		
ffCktCfgTable	ffCktInfo.1		
ffCktEntry	ffCktCfgTable.1	index=*srcport#*	
ffCktSrcNode	ffCktEntry.1	Integer	read only
ffCktSrcIfIndex	ffCktEntry.2	Integer ((*c**1000)+*p*+*t*)	read only
ffCktAdminDestNode	ffCktEntry.9	Integer	
ffCktOperDestNode	ffCktEntry.10	Integer	read only
ffCktAdminDestIfIndex	ffCktEntry.11	Integer ((*c**1000)+*p*+*t*)	
ffCktOperDestIfIndex	ffCktEntry.12	Integer ((*c**1000)+*p*+*t*)	read only
ffCktAdminSrcInsuredRate	ffCktEntry.21	Integer (bits/s) default 0	
ffCktOperSrcInsuredRate	ffCktEntry.22	Integer (bits/s) default -1	read only
ffCktAdminSrcInsuredBurst	ffCktEntry.23	Integer (bytes) default -1	
ffCktOperSrcInsuredBurst	ffCktEntry.24	Integer (bytes) default -1	read only
ffCktAdminSrcMaxRate	ffCktEntry.25	Integer (bits/s) default: -1	
ffCktOperSrcMaxRate	ffCktEntry.26	Integer (bits/s) default: -1	read only
ffCktAdminSrcMaxBurst	ffCktEntry.27	Integer (bytes) default -1	
ffCktOperSrcMaxBurst	ffCktEntry.28	Integer (bytes) default -1	read only
ffCktOperDestInsuredRate	ffCktEntry.30	Integer (bits/s) default -1	read only
ffCktOperDestInsuredBurst	ffCktEntry.32	Integer (bytes) default -1	read only
ffCktOperDestMaxRate	ffCktEntry.34	Integer (bits/s) default: line rate	read only
ffCktOperDestMaxBurst	ffCktEntry.36	Integer (bytes) default -1	read only

ffCktCfgTable Objects	MIB Branch	Value	
ffCktOperPrinBwType	ffCktEntry.37	Integer 1=guaranteed (default) 2=insured	read only
ffCktAdminPrinBwType	ffCktEntry.38	Integer 1=guaranteed (default) 2=insured	
ffCktOperTransPri	ffCktEntry.39	Integer 0, 1 (default)	read only
ffCktAdminTransPri	ffCktEntry.40	Integer 0, 1 (default)	
ffCktStatus	ffCktEntry.99	Integer 1=enable 2=disable (default) 3=delete	

ffCktInfoTable Objects	MIB Branch	Value	
ffCktInfoTable	ffCktInfo.2		
ffCktInfoEntry	ffCktInfoTable.1	index=*port#*	
ffCktInfoIfIndex	ffCktInfoEntry.1	Integer ((*c**1000)+*p*+*t*)	read only
ffCktInfoDownstreamState	ffCktInfoEntry.2	Integer 1=active 2=inactive	read only
ffCktInfoUpstreamState	ffCktInfoentry.3	Integer 1=active 2=inactive	read only
ffCktInfoCallIDIncoming	ffCktInfoEntry.4	Integer (ID)	read only
ffCktInfoCallIDOutgoing	ffCktInfoEntry.5	Integer (ID)	read only
ffCktInfoLastAtmErr	ffCktInfoEntry.6	Octet String	read only
ffCktInfoDataCellsRequired	ffCktInfoEntry.7	Integer (cells)	read only
ffCktInfoLastAtmLocation	ffCktInfoEntry.8	Display String	read only

lsPortProtocols sUniCktInfo Objects	MIB Branch	Value	
sUniCktInfo	lsPortProtocols.5		
sUniCktCfgTable	sUniCktinfo.1		
sUniCktEntry	sUniCktCfgTable.1	index= *srcport#.srcVCI#*	
sUniCktSrcNode	sUniCktEntry.1	Integer	read only
sUniCktSrcIfIndex	sUniCktEntry.2	Integer ((*c**1000)+*p*+*t*)	read only
sUniCktSrcVCI	sUniCktEntry.3	VCI	read only
sUniCktAdminDestNode	sUniCktEntry.10	Integer (ID)	
sUniCktOperDestNode	sUniCktEntry.11	Integer (ID)	read only
sUniCktAdminDestIfIndex	sUniCktEntry.12	Integer ((*c**1000)+*p*+*t*)	

IsPortProtocols sUniCktInfo Objects	MIB Branch	Value	
sUniCktOperDestIfIndex	sUniCktEntry.13	Integer ((*c**1000)+*p*+*t*)	read only
sUniCktAdminDestVCI	sUniCktEntry.14	VCI	
sUniCktOperDestVCI	sUniCktEntry.15	VCI	read only
sUniCktOperPrinBwType	sUniCktEntry.22	Integer 1=guaranteed 2=insured	read only
sUniCktAdminPrinBwType	sUniCktEntry.23	Integer 1=guaranteed 2=insured	
sUniCktOperTransPri	sUniCktEntry.24	Integer (0, 1)	read only
sUniCktAdminTransPri	sUniCktEntry.25	Integer (0, 1)	
sUniCktAdminSrcInsuredRate	sUniCktEntry.26	Integer (cells/s) defaults: MSC: 109 CLC: 218	
sUniCktOperSrcInsuredRate	sUniCktEntry.27	Integer (cells/s) defaults: MSC: 109 CLC: 218	read only
sUniCktAdminSrcInsuredBurst	sUniCktEntry.28	Integer (cells) default 128	
sUniCktOperSrcInsuredBurst	sUniCktEntry.29	Integer (cells) default 128	read only
sUniCktAdminSrcMaxRate	sUniCktEntry.30	Integer (cells/s) default: line rate	
sUniCktOperSrcMaxRate	sUniCktEntry.31	Integer (cells/s) default: line rate	read only
sUniCktAdminSrcMaxBurst	sUniCktEntry.32	Integer (cells) default 128	
sUniCktOperSrcMaxBurst	sUniCktEntry.33	Integer (cells) default 128	read only
sUniCktOperDestInsuredRate	sUniCktEntry.35	Integer (cells/s) defaults: MSC: 109 CLC: 218	read only
sUniCktOperDestInsuredBurst	sUniCktEntry.37	Integer (cells) default 128	read only
sUniCktOperDestMaxRate	sUniCktEntry.39	Integer (cells/s) defaults: MSC: 109 CLC: 218	read only
sUniCktOperDestMaxBurst	sUniCktEntry.41	Integer (cells) default 128	read only
sUniCktAdminSecondaryScale	sUniCktEntry.42	Integer default 255	

lsPortProtocols sUniCktInfo Objects	MIB Branch	Value	
sUniCktOperSecondaryScale	sUniCktEntry.43	Integer default 255	read only
sUniCktSts	sUniCktEntry.99	Integer 1=enable 2=disable (default) 3=delete	

sUniCktInfoTable Objects	MIB Branch	Value	
sUniCktInfoTable	sUniCktInfo.2		
sUniCktInfoEntry	sUniCktInfoTable.1	index=*port#.VCI#*	
sUniCktInfoIfIndex	sUniCktInfoEntry.1	Integer ((*c**1000)+*p*+*t*)	read only
sUniCktInfoVCI	sUniCktInfoEntry.2	VCI	read only
sUniCktInfoUniToNetCallID	sUniCktInfoEntry.3	Integer (ID)	read only
sUniCktInfoNetToUniCallID	sUniCktInfoEntry.4	Integer (ID)	read only
sUniCktInfoUniToNetState	sUniCktInfoEntry.5	Integer 1=active 2=inactive	read only
sUniCktInfoNetToUniState	sUniCktInfoEntry.6	Integer 1=active 2=inactive	read only
sUniCktInfoLastAtmErr	sUniCktInfoEntry.7	Octet String	read only
sUniCktInfoDataCellsRequired	sUniCktInfoEntry.8	Integer (cells)	read only
sUniCktInfoLastAtmLocation	sUniCktInfoEntry.9	Display String	read only

lsPortProtocols sUniCktInfo	MIB Branch	Value	
pvcInfo	lsPortProtocols.6		
pvcCfgTable	pvcinfo.1		
pvcEntry	pvcCfgTable.1	index=*port#.PVC#*	
pvcSrcId	pvcEntry.2	Integer	read only
pvcSrcInsuredRate	pvcEntry.4	Integer ((*c**1000)+*p*+*t*)	
pvcSrcInsuredBurst	pvcEntry.5	VCI	
pvcSrcMaxRate	pvcEntry.6	Integer (bits/s) 0=no commitment default=line rate	
pvcSrcMaxBurst	pvcEntry.7	Integer (bytes) 0=no commitment (default)	
pvcSecondaryScale	pvcEntry.8	Integer 0-100=*x*/100 101-109= (*x*-100)/1000 255 (def.)=use port variable	

lsPortProtocols sUniCktInfo	MIB Branch	Value
pvcPrinBwType	pvcEntry.9	Integer 1=guaranteed (default) 2=insured
pvcTransPri	pvcEntry.10	Integer 0-1 (default 0)
pvcUserDataPerCell	pvcEntry.11	Integer (bits) 0-384 (default 0)
pvcNegotiatedSrcMaxRate	pvcEntry.12	Integer (bits/s) 0=no commitment (default)
pvcStatus	pvcEntry.99	Integer 1=enable 2=disable (default) 3=delete 4=enable multicast

lsPortProtocols sUniCktInfo	MIB Branch	Value	
mcEndptInfo	lsPortProtocols.7		
mcEndptCfgTable	mcEndptinfo.1		
mcEndptEntry	mcEndptCfgTable.1	index= *port#***.***PVC#***.**mcEndptLcl-Instance	
mcEndptLclInstance	mcEndptEntry.3	Integer	read only
mcEndptDest	mcEndptEntry.4	Display String *chassisID***:***port#***:***cktID*	
mcEndptServiceType	mcEndptEntry.5	Integer 1=dest same as src 2=dest FR 3=dest FF 4=dest ATM-UNI 5=dest CBR	
mcEndptRmtVCstatus	mcEndptEntry.6	Integer 1=active 2=inactive	read only
mcEndptCallIDIncoming	mcEndptEntry.7	Integer (ID)	read only
mcEndptDownstreamState	mcEndptEntry.8	Integer 1=active 2=inactive	read only
mcEndptUpstreamState	mcEndptEntry.9	Integer 1=active 2=inactive	read only
mcEndptLastAtmErr	mcEndptEntry.10	Octet String	read only
mcEndptLastAtmLocation	mcEndptEntry.11	Display String *chassID***:***port#*	read only
mcEndptDirection	mcEndptEntry.12	Integer (bits/s) 1=to endpoint 2=from endpoint 3=bidirectional (default)	

lsPortProtocols sUniCktInfo	MIB Branch	Value
mcEndptStatus	mcEndptEntry.99	Integer 1=enable 2=disable 3=delete 4=create request 5=under creation

lsPortProtocols sUniCktInfo	MIB Branch	Value	
pvcCbrInfo	lsPortProtocols.8		
pvcCbrCfgTable	pvcCbrinfo.1		
pvcCbrEntry	pvcCbrCfgTable.1	port#.PVC#	
pvcCbrSrcPvcId	pvcCbrEntry.2	Integer (ID) Ckt emulation or fractional: 1 Crossconnect: dsx1FracIfIndex UNI: *VCI#*	read only
pvcCbrBandwidth	pvcCbrEntry.3	Integer (*cells*/*s*) UNI: configured Ckt emulation or fractional: from port BW (read only) Crossconnect: dsx1FracIfIndex (read only)	 read only read only
pvcCbrAllowableDelay	pvcCbrEntry.4	Integer (ms) default 300000	
pvcCbrClkMode	cemacInfoEntry.5	Integer 1=Adaptive (default) 2=SRTS 3=synchronous 4=other UNI: configurable otherwise: read only	 read only
pvcCbrClkTargetDepth	pvcCbrEntry.6	Integer (bytes)	
pvcCbrClkMaxDepth	pvcCbrEntry.7	Integer	
pvcCbrSliptoFit	pvcCbrEntry.8	Integer (ID) 1=minimize delay (default) 2=minimize slips	
pvcCbrStatus	pvcEntry.99	Integer 1=enable 2=disable (default) 3=delete	

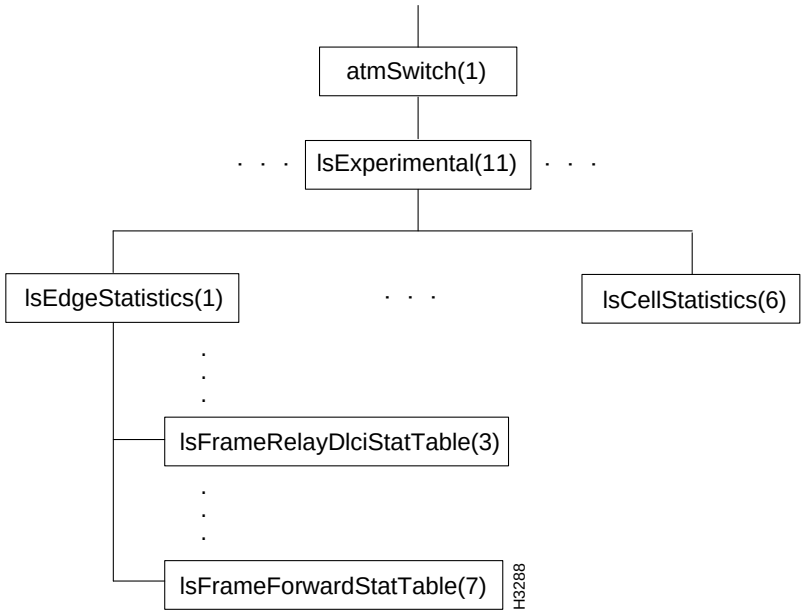
The lsPrivate Branch

The lsPrivate branch of the atmSwitch subtree (atmSwitch.10) is reserved for use in later releases. It is currently empty.

The lsExperimental Branch

The figure below shows portions of the lsExperimental branch of the atmSwitch subtree:

Figure 7-12 The lsExperimental Branch



In Release 2.0, the lsExperimental branch contains objects that are used by Lightstream to provide advanced support functions and analysis of network performance. This branch is not used by most users in Release 2.0. In this section, we describe only those MIB objects in the lsExperimental branch that are used to collect cell statistics and to collect statistics on edge cards for frame relay and frame forwarding. You may use the **getsnmp** command to obtain the statistical information associated with these MIB objects. Ellipses in the figure stand for portions of the lsExperimental branch that are not documented here.

The following tables show the address and type of each object in the selected portions of the branch.

lsFrameRelayDlciStatTable Objects	MIB Branch	Value	
lsEdgeStatistics	lsExperimentalStatistics.1		
lsFrameRelayDlciStatTable	lsEdgeStatistics.3		
lsFrameRelayDlciStatEntry	lsFrameRelayDlciStatTable.1	index= *port#.DLCI#*	
frameRelayDlciStatPortIndex	lsFrameRelayDlciStatEntry.1	Integer	read only
frameRelayDlciStatDlciIndex	lsFrameRelayDlciStatEntry.2	Integer	read only
frameRelayDlciToSwCLP0Frames	lsFrameRelayDlciStatEntry.3	Counter	read only
frameRelayDlciToSwCLP0Cells	lsFrameRelayDlciStatEntry.4	Counter	read only
frameRelayDlciToSwCLP1Frames	lsFrameRelayDlciStatEntry.5	Counter	read only
frameRelayDlciToSwCLP1Cells	lsFrameRelayDlciStatEntry.6	Counter	read only
frameRelayDlciToSwDiscardFrames	lsFrameRelayDlciStatEntry.7	Counter	read only
frameRelayDlciToSwDiscardCells	lsFrameRelayDlciStatEntry.8	Counter	read only
frameRelayDlciFrSwCLP0Frames	lsFrameRelayDlciStatEntry.9	Counter	read only

lsFrameRelayDlciStatTable Objects	MIB Branch	Value	
frameRelayDlciFrSwCLP0Cells	lsFrameRelayDlciStatEntry.10	Counter	read only
frameRelayDlciFrSwCLP1Frames	lsFrameRelayDlciStatEntry.11	Counter	read only
frameRelayDlciFrSwCLP1Cells	lsFrameRelayDlciStatEntry.12	Counter	read only

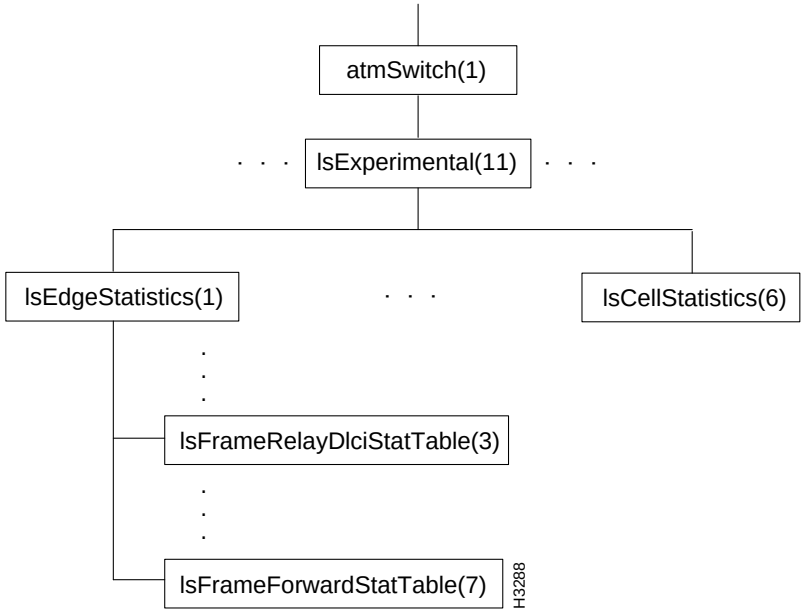
lsFrameForwardStatTable Objects	MIB Branch	Value	
lsEdgeStatistics	lsExperimentalStatistics.1		
lsFrameForwardStatTable	lsEdgeStatistics.7		
lsFrameForwardStatEntry	lsFrameForwardStatTable.1	index=*port#*	
frameForwardStatPortIndex	lsFrameForwardStatEntry.1	Integer	read only
frameForwardToSwCLP0Frames	lsFrameForwardStatEntry.2	Counter	read only
frameForwardToSwCLP0Cells	lsFrameForwardStatEntry.3	Counter	read only
frameForwardToSwCLP1Frames	lsFrameForwardStatEntry.4	Counter	read only
frameForwardToSwCLP1Cells	lsFrameForwardStatEntry.5	Counter	read only
frameForwardToSwDiscardFrames	lsFrameForwardStatEntry.6	Counter	read only
frameForwardToSwDiscardCells	lsFrameForwardStatEntry.7	Counter	read only
frameForwardFrSwCLP0Frames	lsFrameForwardStatEntry.8	Counter	read only
frameForwardFrSwCLP0Cells	lsFrameForwardStatEntry.9	Counter	read only
frameForwardFrSwCLP1Frames	lsFrameForwardStatEntry.10	Counter	read only
frameForwardFrSwCLP1Cells	lsFrameForwardStatEntry.11	Counter	read only

lsCellStatistics Objects	MIB Branch	Value	
lsCellStatistics	lsExperimentalStatistics.6		
lsCellVciStatTable	lsCellStatistics.1		
lsCellVciStatEntry	lsCellVciStatTable.1	index= *port#.VCI#*	
cellVciStatPortIndex	lsCellVciStatEntry.1	Integer	read only
cellVciStatVciIndex	lsCellVciStatEntry.2	Integer	read only
cellVciToSwCLP0Cells	lsCellVciStatEntry.3	Counter	read only
cellVciToSwCLP01Cells	lsCellVciStatEntry.4	Counter	read only
cellVciToSwCLP1Cells	lsCellVciStatEntry.5	Counter	read only
cellVciToSwDiscardCells	lsCellVciStatEntry.6	Counter	read only

The lsIR Branch

The following figure shows the internal routing (lsIR) branch of the atmSwitch subtree:

Figure 7-13 The lsIR Branch



Note The figure does not show all the objects in this branch because there are too many of them. Instead the first and last objects at each level are shown and an ellipsis (...) is used to indicate that additional objects exist. However, all the objects in this branch are listed in the following table.

The table below shows the address and type of each object in the lsIR branch.

Objects in the lsIR Branch	MIB Branch	Value	
irRoutingGroup	lsIR.1		
irRoutingPathsGenerated	irRoutingGroup.1	Counter	read only
irRoutingPathGenSuccess	irRoutingGroup.2	Counter	read only
irRoutingPathGenFailedNoResources	irRoutingGroup.3	Counter	read only
irRoutingPathGenFailedUnknown	irRoutingGroup.4	Counter	read only
irRoutingPathGenFailedOther	irRoutingGroup.5	Counter	read only
irRoutingAveragePathLength	irRoutingGroup.6	Counter	read only

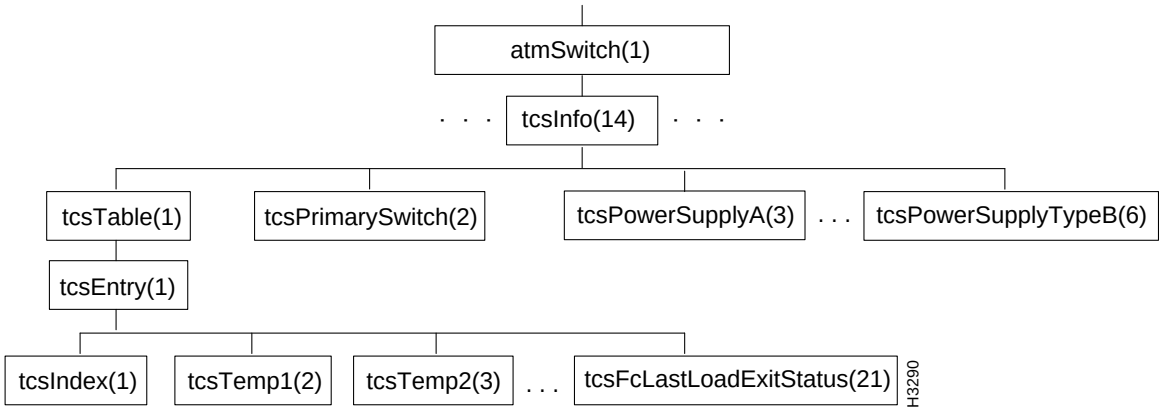
The lsStatistics Branch

In Release 2.1, the lsStatistics branch of the atmSwitch subtree (atmSwitch.13) is empty. It will be used for objects (presently in the lsExperimental branch) that Lightstream uses to analyze network performance.

The tcsInfo Branch

The following figure shows the tcsInfo branch of the atmSwitch subtree:

Figure 7-14 The tcsInfo Branch



Note The figure does not show all the objects in this branch because there are too many of them. Instead the first and last objects at each level are shown and an ellipsis (...) is used to indicate that additional objects exist. However, all the objects in this branch are listed in the following table.

The following table shows the address and type of each object in the branch. The tcsInfo branch contains one table.

Objects in the tcsInfo Branch	MIB Branch	Value	
tcsTable	tcsInfo.1		
tcsEntry	tcsTable.1	index=*card#*	
tcsIndex	tcsEntry.1	Integer (*card#*)	read only
tcsTemp1	tcsEntry.2	Integer	read only
tcsTemp2	tcsEntry.3	Integer	read only
tcsTcsVoltage	tcsEntry.4	Integer	read only
tcsVccVoltage	tcsEntry.5	Integer	read only
tcsScsiVoltage	tcsEntry.7	Integer	read only
tcsPostResult	tcsEntry.8	Octet String	read only

Objects in the tcsInfo Branch	MIB Branch	Value	
tcsCardType	tcsEntry.9	Integer 1=empty 2=error 3=unknown 4=switch 5=NP 6=LS edge 7=LS trunk 8=MS trunk 10=MS edge 11=2-pt FDDI 12=8-pt Ether 13=token ring 14=8-pt LS ser edge 15=8-pt LS ser trunk 30=generic CLC1 31=2-pt OC3 edge 32=2-pt OC3 trunk 33=8-pt T3/E3 edge 34=8-pt T3/E3 trunk 35=2-pt TAXI edge 36=2-pt TAXI trunk 37=8-pt CBR edge 70=switch 2 71=NP switch	read only
tcsPaddleTemp1	tcsEntry.10	Integer	read only
tcsPaddleTemp2	tcsEntry.11	Integer	read only
tcsPaddleWarnTemp1	tcsEntry.12	Integer	read only
tcsPaddleWarnTemp2	tcsEntry.13	Integer	read only
tcsPaddleShutdownTemp1	tcsEntry.14	Integer	read only
tcsPaddleShutdownTemp2	tcsEntry.15	Integer	read only
tcsWarnTemp1	tcsEntry.16	Integer	read only
tcsWarnTemp2	tcsEntry.17	Integer	read only
tcsShutdownTemp1	tcsEntry.18	Integer	read only
tcsShutdownTemp2	tcsEntry.19	Integer	read only
tcsFaultLight	tcsEntry.20	Integer 1=on 2=off	read only
tcsReadyLight	tcsEntry.21	Integer 1=on 2=off	read only
tcsSwitchConnectivityMask	tcsEntry.22	Integer (bitmask) bit 1=*card#* 1, etc.	read only

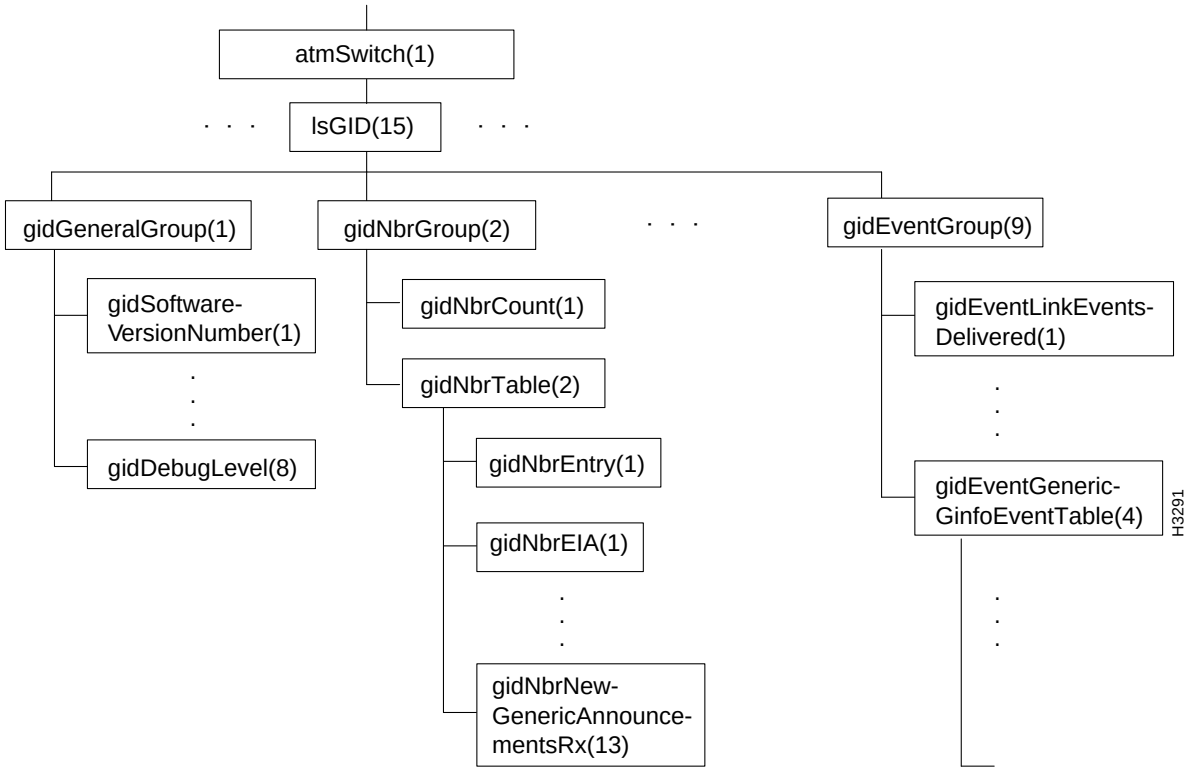
Objects in the tcsInfo Branch	MIB Branch	Value	
tcsFcLoadState	tcsEntry.23	Integer 1=idle 2=request 3=delay 4=delay POST wait 5=loading 6=ND wait 7=succeeded 8=failed 9=too many failures	read only
tcsFcLoadFailedReason	tcsEntry.24	Integer 1=no failure 2=no load flag 3=disabled 4=POST failed 5=fork failed 6=exec failed 7=bad exit 8=bad exit, no reload 9=signalled 10=ND down, no reload 11=ND failed to start 12=not loadable	read only
tcsFcLastLoadExitStatus	tcsEntry.25	Integer	read only
tcsPrimarySwitch	tcsInfo.2	Integer 1=SA 2=SB	
tcsPowerSupplyA	tcsInfo.3	Integer 1=empty 2=failed 3=good	read only
tcsPowerSupplyB	tcsInfo.4	Integer 1=empty 2=failed 3=good	read only
tcsPowerSupplyTypeA	tcsInfo.5	Integer 1=empty 2=DC power tray 3=Todd 4=unknown	read only
tcsPowerSupplyTypeB	tcsInfo.6	Integer 1=empty 2=DC power tray 3=Todd 4=unknown	read only
tcsSwitchFaultMaskA	tcsInfo.7	Integer (bitmask) 0: don’t use switch A *n*: slot *n* clock unhealthy	read only
tcsSwitchFaultMaskB	tcsInfo.8	Integer (bitmask) 0: don’t use switch B *n*: slot *n* clock unhealthy	read only

Objects in the tcsInfo Branch	MIB Branch	Value	
tcsSwitchCutoverSupport	tcsInfo.9	Integer 1=SCs in sync: does lossless 2=SCs not in sync: doesn't do lossless 3=Other: can't do lossless	read only
tcsFCPrimarySwitchA	tcsInfo.10	Integer (bitmask) If bit *n* = 1, Switch A should be in slot *n*.	read only
tcsFCPrimarySwitchB	tcsInfo.11	Integer (bitmask) If bit *n* = 1, Switch B should be in slot *n*.	read only

The lsGID Branch

The figure below shows the IsGID branch of the atmSwitch subtree:

Figure 7-15 The lsGID Branch



Note The figure does not show all the objects in this branch because there are too many of them. Instead, an ellipsis (...) is used to indicate where objects have been omitted from the figure. However, all the objects in this branch are listed in the following tables.

The tables on the following pages show the address and type of each object in the branch.

gidGeneralGroup	MIB Branch	Value	
gidGeneralGroup	lsGID.1		
gidSoftwareVersionNumber	gidGeneralGroup.1	DisplayString	read only
gidProcessID	gidGeneralGroup.2	Integer	read only
gidUpTime	gidGeneralGroup.3	Integer (sec)	read only
gidMemoryUse	gidGeneralGroup.4	Counter (bytes)	read only
gidTimersProcessed	gidGeneralGroup.5	Counter	read only
gidMallocFailures	gidGeneralGroup.6	Counter	read only
gidDebugFlag	gidGeneralGroup.6	Integer (bitmask)	
gidDebugLevel	gidGeneralGroup.6	Integer (0-3) 0=no detail, 3=max	

gidNbrGroup	MIB Branch	Value	
gidNbrGroup	lsGID.2		
gidNbrCount	gidNbrGroup.1	Counter	read only
gidNbrTable	gidNbrGroup.2		
gidNbrEntry	gidNbrTable.1	index=*EIA*	
gidNbrEIA	gidNbrEntry.1	Integer	read only
gidNbrVCI	gidNbrEntry.2	Integer	read only
gidNbrState	gidNbrEntry.3	Integer 1=unknown 2=exists 3=exch. started 4=exchange 5=loading 6=full	read only
gidNbrSyncEvents	gidNbrEntry.4	Counter	read only
gidNbrDBReqListLength	gidNbrEntry.5	Counter	read only
gidNbrDBSumListLength	gidNbrEntry.6	Counter	read only
gidNbrHellosRx	gidNbrEntry.7	Counter	read only
gidNbrLinkAnnouncementsRx	gidNbrEntry.8	Counter	read only
gidNbrNewLinkAnnouncementsRx	gidNbrEntry.9	Counter	read only
gidNbrIPAnnouncementsRx	gidNbrEntry.10	Counter	read only
gidNbrNewIPAnnouncementsRx	gidNbrEntry.11	Counter	read only
gidNbrGenericAnnouncementsRx	gidNbrEntry.12	Counter	read only
gidNbrNewGenericAnnouncementsRx	gidNbrEntry.13	Counter	read only

gidClientGroup	MIB Branch	Value	
gidClientGroup	lsGID.3		
gidClientCount	gidClientGroup.1	Counter	read only

gidClientGroup	MIB Branch	Value	
gidClientTable	gidClientGroup.2		
gidClientEntry	gidClientTable.1	index=gidClientID	
gidClientID	gidClientEntry.1	Integer	read only
gidClientEIA	gidClientEntry.2	Integer	read only
gidClientAnnouncementsRx	gidClientEntry.3	Counter	read only
gidClientLinkAnnouncementsRx	gidClientEntry.4	Counter	read only

gidIOGroup	MIB Branch	Value	
gidIOGroup	lsGID.4		
gidIONbrMsgBuffersFree	gidIOGroup.1	Counter	read only
gidIONbrMsgBuffersActive	gidIOGroup.2	Counter	read only
gidIOClientMsgBuffersFree	idIOGroup.3	Counter	read only
gidIOClientMsgBuffersActive	gidIOGroup.4	Counter	read only
gidClientIPAnnouncementsRx	gidClientEntry.5	Counter	read only
gidClientGenericAnnouncementsRx	gidClientEntry.6	Counter	read only
gidClientEventsTx	gidClientEntry.7	Counter	read only
gidClientPathsGenerated	gidClientEntry.8	Counter	read only

gidSyncGroup	MIB Branch	Value	
gidSyncGroup	lsGID.6		
gidSyncNbrsExistent	gidSyncGroup.1	Counter	read only
gidSyncNbrsExStart	gidSyncGroup.2	Counter	read only
gidSyncNbrsExchange	gidSyncGroup.3	Counter	read only
gidSyncNbrsLoading	gidSyncGroup.4	Counter	read only
gidSyncNbrsFull	gidSyncGroup.5	Counter	read only

gidLinkGroup	MIB Branch	Value	
gidLinkGroup	lsGID.7		
gidLinkDatabaseSize	gidLinkGroup.1	Integer (bytes)	read only
gidLineCardTable	gidLinkGroup.2		
gidLineCardEntry	gidLineCardTable.1	index=*chassisID.card#*	
gidLineCardChassis	gidLineCardEntry.1	Integer (chassis ID)	read only
gidLineCardSlot	gidLineCardEntry.2	Integer	read only
gidLineCardEntryAge	gidLineCardEntry.3	LightStreamUpToMaxAge (seconds)	read only
gidLineCardEntrySeqno	gidLineCardEntry.4	Integer	read only
gidLineCardEntryAdvNP	gidLineCardEntry.5	Integer (NP ID)	read only
gidLineCardPorts	gidLineCardEntry.6	Integer	read only

gidLinkGroup	MIB Branch	Value	
gidPortTable	gidLinkGroup.3		
gidPortEntry	gidPortTable.1	index=*chassisID.port#*	
gidPortChassis	gidPortEntry.1	Integer (chassis ID)	read only
gidPortID	gidPortEntry.2	Integer (*port#*)	read only
gidPortService	gidPortEntry.3	Integer 1=trunk 2=edge	read only
gidPortUpDown	gidPortEntry.4	Integer 1=down 2=up	read only
gidPortBW0	gidPortEntry.5	Integer	read only
gidPortBW1	gidPortEntry.6	Integer	read only
gidPortBW2	gidPortEntry.7	Integer	read only
gidPortRemoteChassis	gidPortEntry.8	Integer (chassis ID)	read only
gidPortRemotePort	idPortEntry.9	Integer (*port#*)	read only

gidIpAddressGroup	Address	Value	
gidIpAddressGroup	lsGID.8		
gidIpAddressDatabaseSize	gidIpAddressGroup.1	Integer (bytes)	read only
gidIpAddressTable	gidIpAddressGroup.2		
gidIpAddressEntry	gidIpAddressTable.1	index= GidInternalIpAddress	
GidInternalIpAddress	gidIpAddressEntry.1	IpAddress	read only
gidIpEntryAge	gidIpAddressEntry.2	LightStreamUpToMaxAge (seconds)	
gidIpEntrySeqno	gidIpAddressEntry.3	Integer	read only
gidIpEntryAdvNP	gidIpAddressEntry.4	Integer (NP ID)	read only
gidIpEntryNetMask	gidIpAddressEntry.5	Integer	read only
gidIpEntryEIA	gidIpAddressEntry.6	Octet String	read only

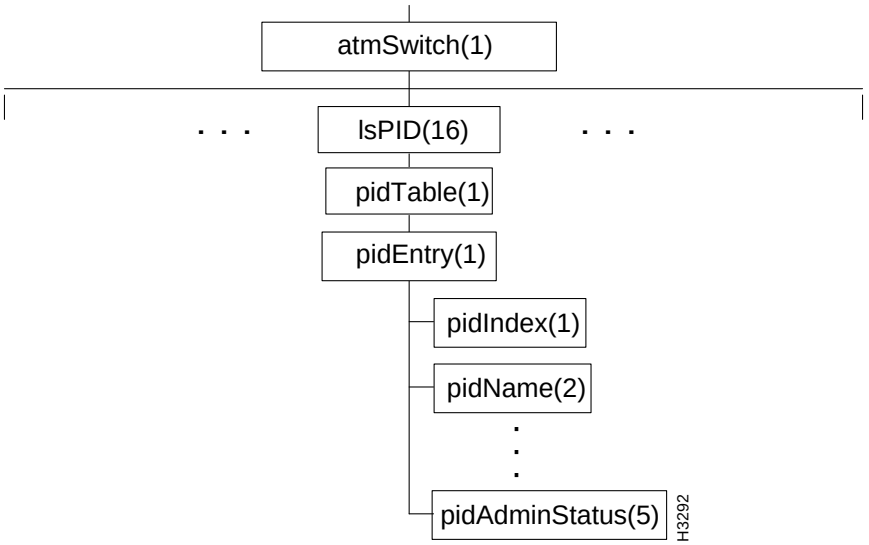
gidEventGroup	MIB Branch	Value	
gidEventGroup	lsGID.9		
gidEventLinkEventsDelivered	gidEventGroup.1	Counter	read only
gidEventIpEventsDelivered	gidEventGroup.2	Counter	read only
gidEventGenericGinfoEventsDelivered	gidEventGroup.3	Counter	read only
gidEventGenericGinfoEventTable	gidEventGroup.4		
gidEventGenericGinfoEventCount	gidEventGeneric-GinfoEventTable.1	index= gidEventDistributionGroup	
gidEventDistributionGroup	gidEventGeneric-GinfoEventCount.1	Integer	read only

gidEventGroup	MIB Branch	Value	
gidEventGenericGinfoEvents	gidEventGeneric-GinfoEventCount.2	Counter	read only

The lsPID Branch

The following figure shows the lsPID branch of the atmSwitch subtree:

Figure 7-16 The lsPID Branch



Note The figure does not show all the objects in this branch because there are too many of them. Instead the first and last objects at each level are shown and an ellipsis (...) is used to indicate that additional objects exist. However, all the objects in this branch are listed in the following table.

The table below shows the address and type of each object in the branch.

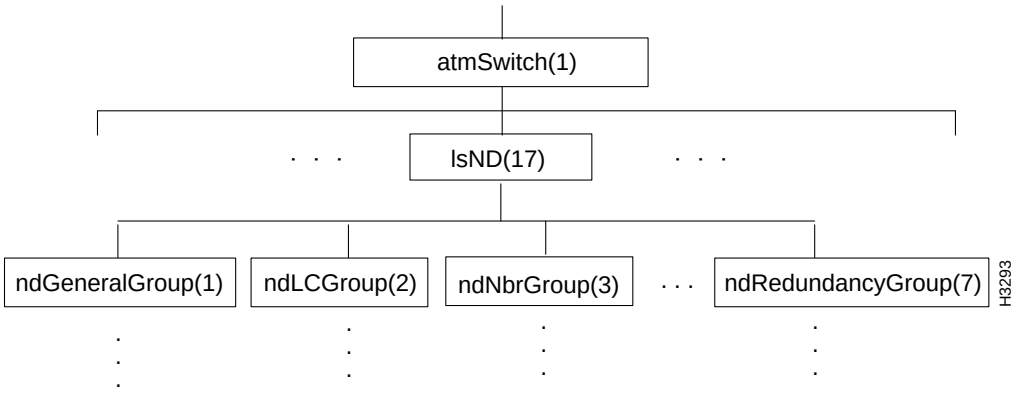
Objects in the lsPID Branch	MIB Branch	Value	
lsPID	atmSwitch.16		
pidTable	lsPID.1		
pidEntry	pidTable.1	index=*PID#*	
pidIndex	pidEntry.1 (*PID#*)	Integer	read only
pidName	pidEntry.2	Integer	read only
pidCreationTime	pidEntry.3	Integer (sec)	read only
pidOperStatus	pidEntry.4	Integer 1=up 2=down	read only

Objects in the lsPID Branch	MIB Branch	Value
pidAdminStatus	pidEntry.5	Integer 1=up 2=down

The lsND Branch

The figure below shows the lsND branch of the atmSwitch subtree:

Figure 7-17 The lsND Branch



Note The figure does not show all the objects in this branch because there are too many of them. Instead the first and last objects at each level are shown and an ellipsis (...) is used to indicate that additional objects exist. However, all the objects in this branch are listed in the following tables.

The following tables show the address and type of each object in the branch.

ndGeneralGroup	MIB Branch	Value	
ndGeneralGroup	lsND.1		
ndSoftwareVersionNumber	ndGeneralGroup.1	Display String	read only
ndProcessID	ndGeneralGroup.2	Integer (*PID#*)	read only
ndMemoryUse	ndGeneralGroup.4	Counter (bytes)	read only
ndTimersProcessed	ndGeneralGroup.5	Counter	read only

ndLCGroup	MIB Branch	Value	
ndLCGroup	lsND.2		
ndLCCount	ncLCGroup.1	Counter	read only
ndLCTable	ncLCGroup.2		
ncLCEntry	ndLCTable.1	index=*EIA*	
ndLCEIA	ndLCEntry.1	Integer	read only
ndLCChannel	ndLCEntry.2	Integer	read only

ndLCGroup	MIB Branch	Value	
ndLCState	ndLCEntry.3	Integer 1=unknown 2=exists 3=up 4=coming down	read only

ndNbrGroup	MIB Branch	Value	
ndNbrGroup	lsND.3		
ndNbrCount	ndNbrGroup.1	Counter	read only
ndNbrTable	ndNbrGroup.2		
ndNbrEntry	ndBbrTable.1	index=*EIA*	
ndNbrEIA	ndNbrEntry.1	Integer	read only
ndNbrChannel	ndNbrEntry.2	Integer	read only
ndNbrState	ndNbrEntry.3	Integer 1=unknown 2=exists 3=up 4=coming down	read only

ndSwudGroup	MIB Branch	Value	
ndSwudGroup	lsND.4		
ndSwudTable	ndSwudGroup.1		
ndSwudEntry	ndSwudTable.1	index=ndSwudIndex	
ndSwudIndex	ndSwudEntry.1	Integer (*slot#*)	read only
ndOperIntvl	ndSwudEntry.2	Integer (*x**100ms)	read only
ndOperJ	ndSwudEntry.3	Integer	read only
ndOperK	ndSwudEntry.4	Integer	read only
ndOperM	ndSwudEntry.5	Integer	read only
ndOperN	ndSwudEntry.6	Integer	read only
ndAdminIntvl	ndSwudEntry.7	Integer (*x**100ms)	
ndAdminJ	ndSwudEntry.8	Integer	
ndAdminK	ndSwudEntry.9	Integer	
ndAdminM	ndSwudEntry.10	Integer	
ndAdminN	ndSwudEntry.11	Integer	
ndTrigger	ndSwudEntry.12	Integer	
ndSwudStatsInputErrors	ndSwudGroup.2	Counter	read only
ndSwudStatsTable	ndSwudGroup.3		
NdSwudStatsEntry	ndSwudStatsTable.1	index=ndSwudStatsIndex	
ndSwudStatsIndex	ndSwudStatsEntry.1	Integer	read only
ndInputCells	ndSwudStatsEntry.2	Counter	read only
ndInputErrs	ndSwudStatsEntry.3	Counter	read only

ndSwudGroup	MIB Branch	Value	
ndOutputCells	ndSwudStatsEntry.4	Counter	read only
ndOutputErrs	ndSwudStatsEntry.5	Counter	read only

ndClient	MIB Branch	Value	
ndClient group	lsND.5		
ndClientCount	ndClientGroup.1	Counter	read only
ndClientTable	ndClientGroup.2		
ndClientEntry	ndClientTable.1	index=ndClientID	
ndClientID	ndClientEntry.1	Integer	read only
ndClientType	ndClientEntry.2	Integer 3=ND 4=GID 5=LCC 6=CA 7=Sys	read only
ndClientSubType	ndClientEntry.3	Integer	read only
ndClientEIA	ndClientEntry.4	Integer	read only
ndClientRegistration	ndClientEntry.5	Integer	read only

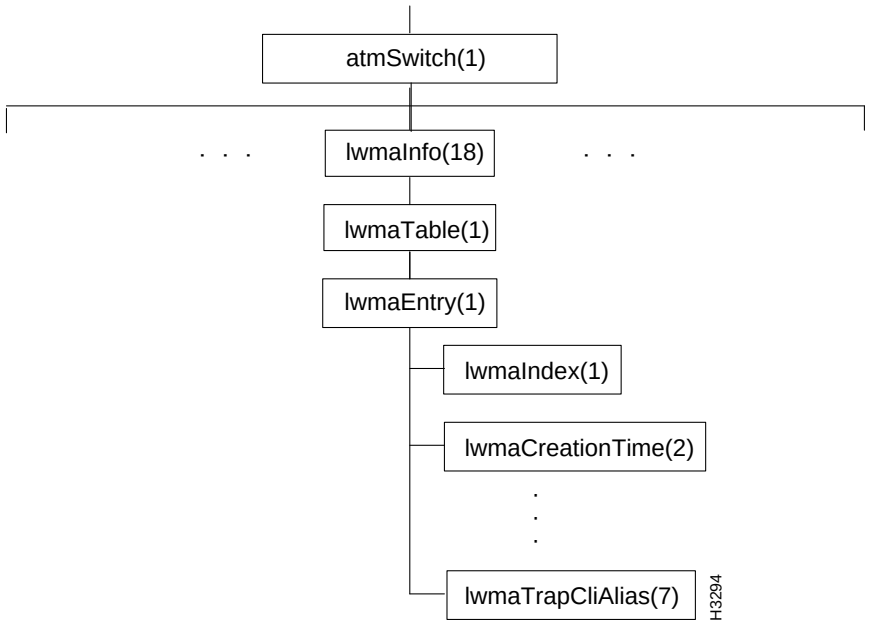
ndInternalGroup	MIB Branch	Value
ndInternalGroup	lsND.6	
ndInternalDebugLevel	ndInternalGroup.1	Integer 0=off (default)
ndInternalDebugFlags	ndInternalGroup.2	Integer 0=none (default)

ndRedundancyGroup	MIB Branch	Value	
ndRedundancyGroup	lsND.7		
ndPrimaryNP	ndRedundancyGroup.1	Integer (*slot#*) 0=none (default)	read only
ndThisNP	ndRedundancyGroup.2	Integer (*slot#*)	read only
ndForceToBackup	ndRedundancyGroup.3	Integer	

The lwmaInfo Branch

The following figure shows the lwmaInfo branch of the atmSwitch subtree:

Figure 7-18 The lwmaInfo Branch



Note The figure does not show all the objects in this branch because there are too many of them. Instead the first and last objects at each level are shown and an ellipsis (...) is used to indicate that additional objects exist. However, all the objects in this branch are listed in the following table.

The following table shows the address and type of each object in the branch.

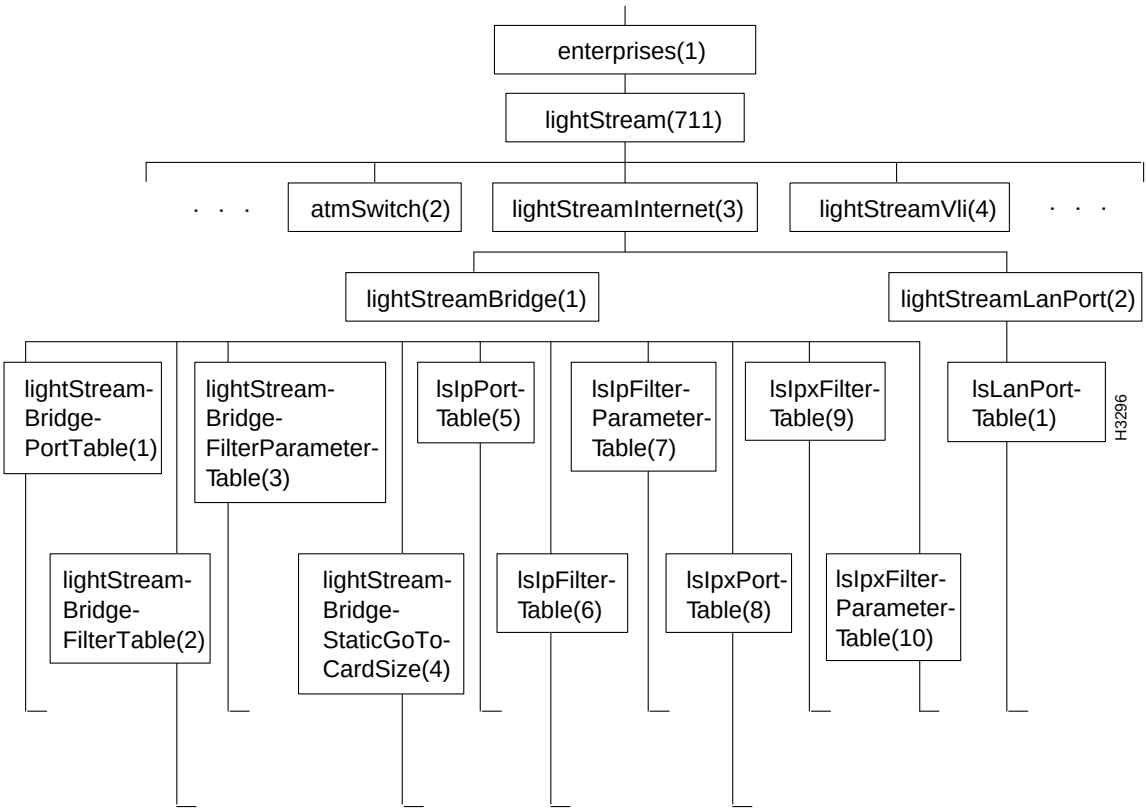
Objects in the lwmaInfo Branch	MIB Branch	Value	
lwmaTable	lwmaInfo.1		
lwmaEntry	lwmaTable.1	index=*LWMA#*	
lwmaIndex	lwmaEntry.1	Integer (LWMA #)	read only
lwmaCreationTime	lwmaEntry.2	Integer (seconds)	read only
lwmaTableNotification	lwmaEntry.3	Object Identifier	read only
lwmaTrapLevel	lwmaEntry.4	Integer 1=operational 2=informational 3=trace 4=debug	
lwmaTrapNumber	lwmaEntry.5	Integer	
lwmaTrapOnOffState	lwmaEntry.6	Integer 1=on 2=off	

Objects in the lwmaInfo Branch	MIB Branch	Value
lwmaTrapCliAlias	lwmaEntry.7	Octet String

The lightStreamInternet Subtree

The following figure shows the lightStreamBridge and lightStreamLanPort branches in the lightStreamInternet subtree of the LightStream private MIB:

Figure 7-19 The lightStreamInternet Subtree



Note The figure does not show all the objects in these branches because there are too many of them. Instead the first and last objects at each level are shown and an ellipsis (...) is used to indicate that additional objects exist. However, all the objects in these branches are listed in the following tables.

The following tables show the address and type of each object in the lightStreamBridge and lightStreamLanPort branches.

lightStreamBridgePortTable	MIB Branch	Value
lightStreamBridgePortTable	lightStreamBridge.1	
lightStreamBridgePortEntry	lightStreamBridgePortTable.1	index=*port#*

lightStreamBridgePortTable	MIB Branch	Value	
lightStreamBrPortPort	lightStreamBridgePortEntry.1	Integer (dot1dBasePort)	read only
lightStreamBrPortDefaultAction	lightStreamBridgePortEntry.2	1 = forward (def.) 2 = block	
lightStreamBrPortBcastRateLimit	lightStreamBridgePortEntry.3	Integer (packets) 1=forward all (def.)	
lightStreamBrPortDroppedBcastPkts	lightStreamBridgePortEntry.4	Counter (packets)	read only

lightStreamBridgeFilterTable	MIB Branch	Value	
lightStreamBridgeFilterTable	lightStreamBridge.2		
lightStreamBridgeFilterEntry	lightStreamBridgeFilterTable.1	index= *ID.TokenIndex*	
lightStreamBrFilterId	lightStreamBridgeFilterEntry.1	Integer	read only
lightStreamBrFilterTokenIndex	lightStreamBridgeFilterEntry.2	Integer	read only
lightStreamBrFilterTokenType	lightStreamBridge	Integer 1=operation 2=Frame Field 3=MAC address 4=Ether 5=LLC SAP 6 (reserved) 7=SNAP OUI	
lightStreamBrFilterTokenValue	lightStreamBridgeFilterEntry.4	DisplayString	
lightStreamBrFilterStatus	lightStreamBridgeFilterEntry.5	Integer 1=complete 2=delete 4=intermediate token	

lightStreamBridgeFilterParameter-Table	MIB Branch	Value	
lightStreamBridgeFilterParameterTable	lightStreamBridge.3		
lightStreamBridgeFilterParameterEntry	lightStreamBridgeFilter-ParameterTable.1	index= *port#.FilterId*	
lightStreamBrFilterParmPort	lightStreamBridgeFilter-ParameterEntry.1	Integer (dot1dBasePort)	read only
lightStreamBrFilterParmFilterId	lightStreamBridgeFilter-ParameterEntry.2	Integer	read only
lightStreamBrFilterParmFilterPriority	lightStreamBridgeFilter-ParameterEntry.3	Integer 1=highest (default largest + 10)	
lightStreamBrFilterParmAction	lightStreamBridgeFilter-ParameterEntry.4	LightStreamFilterAction 1 = forward 2 = block	
lightStreamBrFilterParmMatchCounts	lightStreamBridgeFilter-ParameterEntry.5	Counter	read only

lightStreamBridgeFilterParameter-Table	MIB Branch	Value
lightStreamBrFilterParmValidation	lightStreamBridgeFilter-ParameterEntry.6	Integer 1=valid 2=invalid
lightStreamBrFilterParmTp	lightStreamBridgeFilter-ParameterEntry.7	Integer
lightStreamBrFilterParmMg	lightStreamBridgeFilter-ParameterEntry.8	Integer

lightStreamBridgePortTable	MIB Branch	Value	
lightStreamBridgeStaticGoToCardSize	lightStreamBridge.4	Integer (bytes of dot1dStaticGoTo per card)	read only

lightStreamBridgeFilter-ParameterTable	MIB Branch	Value
lsIpPortTable	lightStreamBridge.5	
lsIpPortEntry	lsIpPortTable.1	index=*port#*
lsIpPortDefaultAction	lsIpPortEntry.2	LightStreamFilterAction 1 = forward (default) 2 = block

lightStreamBridgeFilterTable	MIB Branch	Value	
lsIpFilterTable	lightStreamBridge.6		
lsIpFilterEntry	lsIpFilterTable.1	index=*ID.TokenIndex*	
lsIpFilterId	lsIpFilterEntry.1	Integer	read only
lsIpFilterTokenIndex	lsIpFilterEntry.2	Integer	read only
lsIpFilterTokenType	lsIpFilterEntry.3	Integer 1=operation 2=packet field 3=integer value 4=IP address	
lsIpFilterTokenValue	lsIpFilterEntry.4	DisplayString	
lsIpFilterStatus	lsIpFilterEntry.5	Integer 1=complete 2=delete 4=intermediate token	

lightStreamBridgeFilter-ParameterTable	MIB Branch	Value	
lsIpFilterParameterTable	lightStreamBridge.7		
lsIpFilterParameterEntry	lsIpFilterParameterTable.1	index=*port#.FilterID*	
lsIpFilterParmFilterId	lsIpFilterParameterEntry.2	Integer	read only

lightStreamBridgeFilter-ParameterTable	MIB Branch	Value	
lsIpFilterParmFilterPriority	lsIpFilterParameterEntry.3	Integer	
lsIpFilterParmAction	lsIpFilterParameterEntry.4	LightStreamFilterAction 1 = forward 2 = block	
lsIpFilterParmMatchCounts	lsIpFilterParameterEntry.5	Counter	read only
lsIpFilterParmValidation	lsIpFilterParameterEntry.6	Integer 1=valid 2=invalid	
lsIpFilterParmTp	lsIpFilterParameterEntry.7	Integer	
lsIpFilterParmMg	lsIpFilterParameterEntry.8	Integer	

lightStreamBridgeFilterParameterTable	MIB Branch	Value
lsIpxPortTable	lightStreamBridge.8	
lsIpxPortEntry	lsIpxPortTable.1	index=*port#*
lsIpxPortDefaultAction	lsIpxPortEntry.2	LightStreamFilterAction 1 = forward (default) 2 = block

lightStreamBridgeFilterTable	MIB Branch	Value	
lsIpxFilterTable	lightStreamBridge.9		
lsIpxFilterEntry	lsIpxFilterTable.1	index=*ID.TokenIndex*	
lsIpxFilterId	lsIpxFilterEntry.1	Integer	read only
lsIpxFilterTokenIndex	lsIpxFilterEntry.2	Integer	read only
lsIpxFilterTokenType	lsIpxFilterEntry.3	Integer 1=operation 2=packet field 3=packet type 5=network 6=node 7=socket	
lsIpxFilterTokenValue	lsIpxFilterEntry.4	Display String	
lsIpxFilterStatus	lsIpxFilterEntry.5	Integer 1=complete 2=delete 4=intermediate token	

lightStreamBridgeFilter-ParameterTable	MIB Branch	Value	
lsIpxFilterParameterTable	lightStreamBridge.10		
lsIpxFilterParameterEntry	lsIpxFilterParameterTable.1	index=*port#.FilterID*	
lsIpxFilterParmFilterId	lsIpxFilterParameterEntry.2	Integer	read only
lsIpxFilterParmFilterPriority	lsIpxFilterParameterEntry.3	Integer	

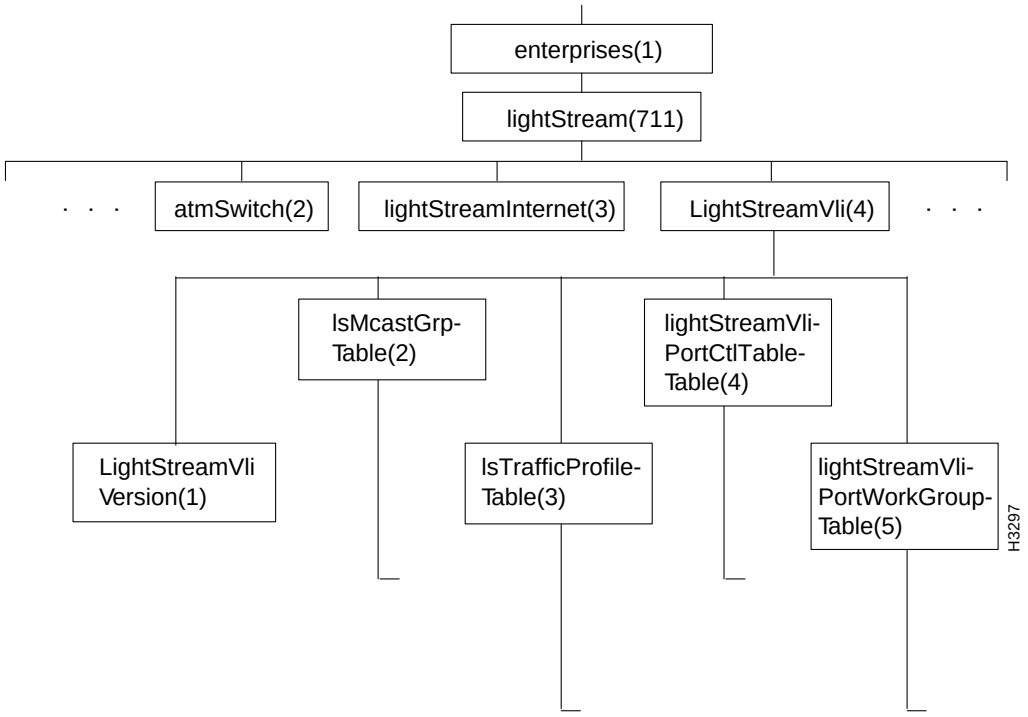
lightStreamBridgeFilter-ParameterTable	MIB Branch	Value	
lsIpxFilterParmAction	lsIpxFilterParameterEntry.4	LightStreamFilterAction 1=forward 2=block	
lsIpxFilterParmMatchCounts	lsIpxFilterParameterEntry.5	Counter	read only
lsIpxFilterParmValidation	lsIpxFilterParameterEntry.6	Integer 1=valid 2=invalid	
lsIpxFilterParmTp	lsIpxFilterParameterEntry.7	Integer	
lsIpxFilterParmMg	lsIpxFilterParameterEntry.8	Integer	

sLanPortTable Objects	MIB Branch	Value
lsLanPortTable	lightStreamLanPort.1	
lsLanPortEntry	lsLanPortTable.1	index=*port#*
lsLanPortNpTrafficFilter	lsLanPortEntry.1	Integer 1=forward 2=block (default)

The lightStreamVLI Subtree

The following figure shows the lightStreamVli subtree in the LightStream private MIB:

Figure 7-20 The lightStreamVLI Subtree



The following tables describe the objects in the lightStreamVLI subtree.

lightStreamVLI Version	MIB Branch	Value	
lightStreamVliVersion	lightStreamVli.1	Integer 1=Version 1	read only

lightStreamBridgeFilterTable	MIB Branch	Value	
lsMcastGrpTable	lightStreamVli.2		
lsMcastGrpEntry	lsMcastGrpTable.1	index= *GrpID.chassisID.port#*	
lsMcastGrpId	lsMcastGrpEntry.1	Integer	read only
lsMcastChassisId	lsMcastGrpEntry.2	Integer	read only
lsMcastIfIndex	lsMcastGrpEntry.3	Integer	read only
lsMcastGrpEntryValidation	lsMcastGrpEntry.4	Integer 1=valid 2=invalid	

lsTrafficProfileTable	MIB Branch	Value	
lsTrafficProfileTable	lightStreamVli.3		
lsTrafficProfileEntry	lsTrafficProfile- Entry.1	index=lsTpId (tprof ID)	
lsTpId	lsTrafficProfile- Entry.1	Integer	read only
lsTpInsuredRate	lsTrafficProfile- Entry.2	Integer (bits) 0-100,000,000 (default 0)	
lsTpInsuredBurst	lsTrafficProfile- Entry.3	Integer (bytes/s) 0-64,000 (default 0)	
lsTpMaxRate	lsTrafficProfile- Entry.4	Integer (bits) 64,000-100,000,000 -1 = default	
lsTpMaxBurst	lsTrafficProfile- Entry.5	Integer (bytes/s) 0-64,000 (default 32,000)	
lsTpSecondaryScale	lsTrafficProfile- Entry.6	Integer (0-109, default 1)	
lsTpPrinBwType	lsTrafficProfile- Entry.7	Integer 1=Guaranteed 2=Insured (default)	
lsTpTransPri	lsTrafficProfile- Entry.8	Integer 0=low (default) 1=high	
lsTpValidation	lsTrafficProfile- Entry.9	Integer 1=valid 2=invalid	

lightStreamVliPortCtlTable	MIB Branch	Value
lightStreamVliPortCtlTable	lightStreamVli.4	
lightStreamVliPortCtlEntry	lightStreamVliPortCtlTable.1	index=*port#*

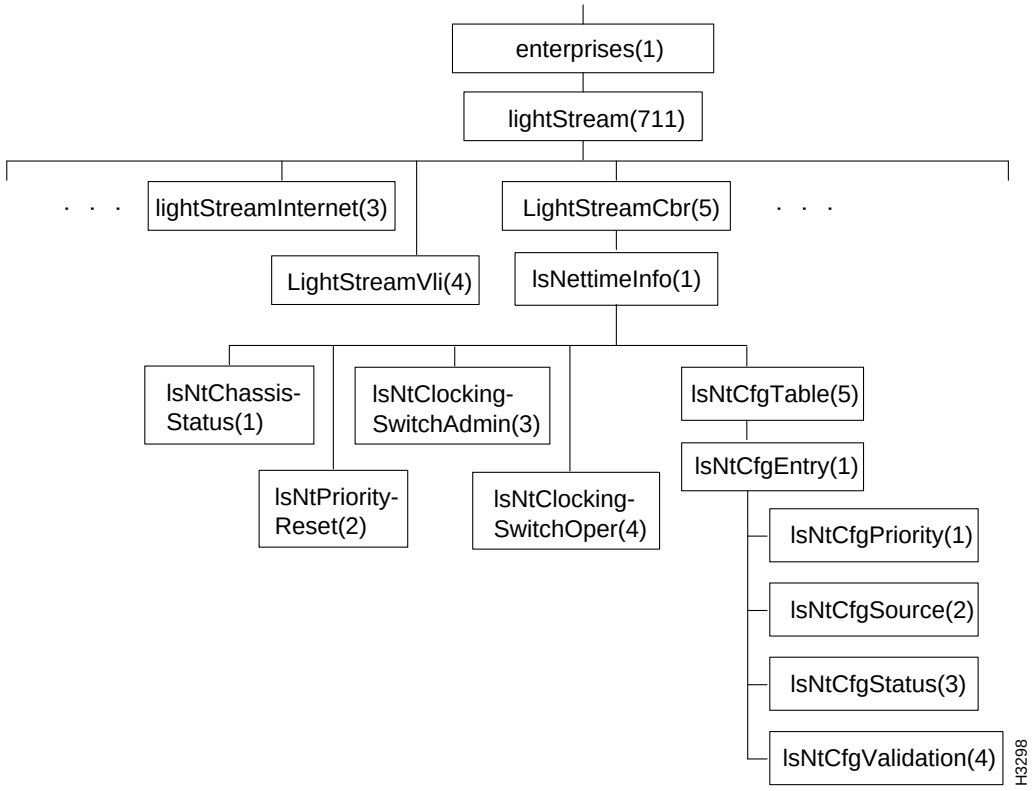
lightStreamVliPortCtlTable	MIB Branch	Value	
lightStreamVliPortCtlPort	lightStreamVliPortCtlEntry.1	Integer (dot1dBasePort)	read only
lightStreamVliPortCtlMode	lightStreamVliPortCtlEntry.2	Integer 1=include (default) 2=exclude	

lightStreamVliPortWorkGroup	MIB Branch	Value	
lightStreamVliPortWorkGroupTable	lightStreamVli.5		
lightStreamVliPortWorkGroupEntry	lightStreamVliPort-WorkGroupTable.1	index=*port#.wgrpID*	
lightStreamVliPortWorkGroupPort	lightStreamVliPort-WorkGroupEntry.1	Integer (dot1dBasePort)	read only
lightStreamVliPortWorkGroupID	lightStreamVliPort-WorkGroupEntry.2	Integer (0-65535)	read only
lightStreamVliPortWorkGroup-Validation	lightStreamVliPort-WorkGroupEntry.3	Integer 1=valid 2=invalid	

The lightStreamCbr Subtree

The following figure shows the lightStreamCbr subtree in the LightStream private MIB:

Figure 7-21 The lightStreamCbr Subtree



The following table describes the objects in the lightStreamCbr subtree.

lightStreamVliPortWorkGroup	MIB Branch	Value	
lsNettimeInfo	lightStreamCbr.1		
lsNtChassisStatus	lsNettimeInfo.1	Integer (lsNtCfgPriority) 0=no nettime clocking	read only
lsNtPriorityReset	lsNettimeInfo.2	Integer 1-10 (1 default)	
lsNtClockingSwitchAdmin	lsNettimeInfo.3	Integer 1=switch A 2=switch B 3=autoselect (default)	
lsNtClockingSwitchOper	lsNettimeInfo.4	Integer 1=switch A 2=switch B 3=autoselect	read only
lsNtCfgTable	lsNettimeInfo.5		
lsNtCfgEntry	lsNtCfgTable.1	index=lsNtCfgPriority	
lsNtCfgPriority	lsNtCfgEntry.1	Integer 1-10 Defaults: 1 for BITs plug 2 for oscillator	read only
lsNtCfgSource	lsNtCfgEntry.2	Integer 1=BITs plug 2=switch card oscillator ifIndex =physical interface	
lsNtCfgStatus	lsNtCfgEntry.3	Integer 1=clock up 2=clock down 3=no longer available	read only
lsNtCfgValidation	lsNtCfgEntry.4	Integer 1=valid 2=invalid	