Ascend Enterprise MIB Definitions

Ascend Communications, Inc.

Product Code: 80015
Revision 02
March 1997

NOTICE

This manual is supplied without representation or warranty of any kind. Cascade Communications Corp. assumes no responsibility and shall have no liability of any kind arising from supply or use of this publication or any material contained herein.

The information contained herein is proprietary to Cascade Communications Corp. and/or its vendors, and its use or disclosure is subject to the restrictions stated in the standard Cascade Communications Corp. license terms and conditions or the appropriate third-party sublicense agreement.

Cascade Enterprise MIB Definitions and the Cascade logo are trademarks of Cascade Communications Corp.

All other product names are trademarks or registered trademarks of their respective companies.

Mention of third-party products is for informational purposes only and constitutes neither an endorsement nor a recommendation. Cascade Communications Corp. assumes no responsibility with regard to the performance of these products.

Cascade Communications Corp.
5 Carlisle Road
Westford, MA 01886

Contents

How To Read The Ascend MIB

This document lists the variables in the Ascend MIB database, and provides instructions on how to determine the command you need to enter to find information about a given component on the network.

About MIBs

When you want to access a specific variable from a MIB group, you type a command which uses the following format:

{group name}.{Table}.{Entry}.{Entry #}.{INDEX}

For example, if want to know a card's Admin Status: whether a card is up, down, testing, or in maintenance mode, use the following steps:

1. Find the beginning of the Card Group in the MIB document. This gives you the group name: group name = Card

```
The Card Group
--  The variables that manage intelligent cards (PP's, CP's, IOP's).
--  For redundant pairs of cards, only the active card is managed.
```

2. Find the number associated with the Table index, **cardTable**. Look at the line, **::={card 2}**. The Table value = 2

```
cardTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF CardEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A list of cardEntry's. The number of entries is given by
                the value of cardNumber"
        ::= { card 2}
```

3. Find the number associated with the Entry index, **cardEntry**.Look at the line,**::={cardTable 1}**. The Entry index = 1.

```
cardEntry OBJECT-TYPE
        SYNTAX  CardEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "The card entry contains objects relevant to managing
                intelligent cards."
        INDEX       { cardPhysicalSlotId, cardRedundState }
        ::= { cardTable 1}
```

4. Find Entry # for the MIB variable you want to access. To retrieve a card's Admin Status, you need to access the variable, **cardAdminStatus**. Look at the line, **::={cardEntry 6}**. The Entry # value = 6.

```
cardAdminStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                    invalid (0),
                    up (1),
                    down (2),
                    testing (3),
                    maintenance (6)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The desired status of this card."
        ::= { cardEntry 6}
```

5. With the information you know at this point, the command you would begin to enter is:

GET CARD.2.1.6.

You know the values for Table (2) and Entry (1) for a card, and the Entry # (6) for the MIB variable.

6. Find the INDEX items you need to provide to complete this command. The **cardEntry** variable you looked at in step 3 contains the INDEX items.

```
INDEX       { cardPhysicalSlotId, cardRedundState }
```

For this example, the card you are requesting information for is in slot 4. The **card PhysicalSlotId = 4**. It is configured as the primary card, not redundant. The **cardRedundState = 1**. Therefore, the INDEX = 4.1.

7. Type the following command to retrieve the Admin Status for a card:

GET CARD.2.1.6.4.1

where, {group name=CARD}.{Table=2}.{Entry=1}.{Entry #=6}.{INDEX=4.1}

The system responds by displaying the command as the full MIB tree index, 1.3.1.3.1.277.2.1.6.4.1, and retrieves an integer which reflects the card's Admin Status. Refer to the **cardAdminStatus** variable to interpret the this integer. For this example,

```
SYNTAX  INTEGER {
                    invalid (0),
                    up (1),
                    down (2),
                    testing (3),
                    maintenance (6)
```

If the command returns a value of 1, the Admin Status of the card is up; if it returns a value of 2, the card is down.

Variable Groups

There are ten groups of general interest in the Cascade MIB. Each one of these groups has a list of MIB variables associated with it. You can type a shortcut for the group name to access the information tracked by these variables.

The shortcut for some group names, such as card and pport use the format {groupname}. {Table}.{Entry}. The shortcut provides the Table and Entry values, for example, **Card.2.1**. You only need to specify Entry # and INDEX.

The group names, node, net, ip, ase, and system do not use Table and Entry values. Their shortcut uses the format {groupname}.(Entry#}.0. For example, to retrieve the IP network number, look in the MIB at the group name **Net**. Look for the MIB variable, **netNumber**.The Entry # is **::={net 2}**. Enter the following command: **get net.2.0**.

Table 1 lists these variable groups, along with the shortcut and indexing scheme for each group.

Table 1: MIB Variable Group Descriptions

Group Name	Description	Shortcut	Index
Node	Information about the switch at the node level.	node.	0
Net	Information about the switch at the network level	net.	0
IP	Information about how the switch sees the routes to the rest of the network and the NMS.	ip.	0
ASE	Information about how the switch communi-cates with the NMS if it is a gateway switch.	ase.	0
System	High level informa-tion about the switch.	system.	0
Card	Information about each of the cards in the switch.	card.2.1.	slot number, redundancy status

Table 1: MIB Variable Group Descriptions

Group Name	Description	Shortcut	Index
Pport	Information about each of the physical ports in the switch.	pport.2.1.	slot number, pport number
Lport	Information about the configuration of each logical port in the switch.	lport.1.1.	interface number
Interface	Information about the statistics for each logical port in the switch.	interface.2.1.	interface index
ckt	Information about each of the PVCs in the switch.	ckt.1.1.	interface number, DLCI

SNMP Commands

You can use the following direct snmp commands in debug mode:

GET
Retrieves the status of a specific MIB variable. For example, to retrieve the operational status of a physical port, enter the following command:

GET PPORT.2.1.9.Y.Z

Where Y = the slot number and Z = the port number. To interpret the status that this command retrieves, review the SYNTAX INTEGER for the operational status variable, **pportOperStatus**

NEXT
Retrieves the status for all physical ports using the MIB variable you specify. For example, to retrieve the Admin status of all physical ports, enter the following command:

NEXT PPORT.2.1.9

Where 9 = the Entry # of the MIB variable, **pportAdminStatus**

This command retrieves information for all physical ports, according to the MIB variable you designate. (**Note:** Expect a lengthy printout of information!)

SET
Modifies the current status of a specific MIB variable. For example, to modify the operational status of a physical port, enter the following command:

SET PPORT.2.1.9.Y.Z. Q

Where Y = the slot number, Z = the port number, and Q = the operational status you want to set. To find the SYNTAX INTEGER that represents the operational status, look up the variable, **pportOperStatus**, in the MIB.

More Examples

get node.4.0
This command retrieves the state of the switch. It uses the shortcut **{groupname}.{Entry #}.0**. Remember, the group names: node, ip, ase, net, and system, all use this shortcut. To access the state of the switch, look up the MIB variable, **nodeState**. The Entry # for this variable is :: = {node **4**}.

The command retrieves one of the following values which reflects the state of the switch:

1=down
2=initializing
3=active
4=marginal
5=testing

get pport.2.1.9.y.z
This command retrieves the administrative status of the physical port you specify. It uses the shortcut from Table 1, **pport.2.1**. To access the Admin Status of the physical port, look up the MIB variable, **pportAdminStatus**. The Entry # for this variable is :: = {pportEntry **9** }.

To find out what INDEX values (**x** and **y**) you must provide to complete any physical port group command, look up the MIB variable, **pportEntry**. This variable specifies an INDEX of slot number (**y**) and physical port number (**z**).

The command retrieves one of the following values which reflects the Admin status of the physical port:

0=invalid
1=up
2=down
3=testing

get lport.1.1.22.x
This command retrieves the IP address of the node at the other end of a trunk. It uses the shortcut from Table 1, **lport.1.1**. To access the IP address of this node, look up the MIB variable, **lportTrkRnode**. The Entry # for this variable is :: = {lportEntry **22**}.

To find out what INDEX value (**x**) you must provide to complete any logical port group command, look up the MIB variable, **lportEntry**. This variable specifies an INDEX of lportIfIndex, meaning, you must supply the interface number (ifnum) of the logical port.

get ckt.1.1.7.y.z
This command retrieves the IP address of the node at the other end of the circuit. It uses the shortcut from Table 1, **ckt.1.1**. To access the IP address of this node, look up the MIB variable, **cktDestNodeId**. The Entry # for this variable is :: = {cktEntry **7**}.

To find out what INDEX values (**y** and **z**) you must provide to complete any circuit group command, look up the MIB variable, **cktEntry**. This variable specifies an INDEX of interface number (**y**) and DLCI number (**z**).

Cascade Communications STDX/B-STDX MIB Definitions

```
-- Date      Engineer           Description
-- ------------------------------------------------------------------------
---
-- 05/11/94  Mike Bernstein     Added new pportInterface types for E1
-- 06/10/94  Ren Yonghong       Added some SVC objects
-- 10/06/94  Ren Yonghong       Reserved trap number 36 for SVC
-- 10/06/94  Arvind Puntambekar Added SMDS modifications for 4.0 release
-- 10/20/94  Ren Yonghong       Detailed all possible values for
--                                 pportDataRate
-- 11/15/94  Ren Yonghong       Clarified and enhanced several objects
-- 12/14/94  Dan Bergman        Added ISDN objects
-- 12/15/94  Chris D'Souza      Added dsx3 loopback objects
-- 12/21/94  Eric Li            Added dsx3 FEAC status object
-- 01/09/95  Mike Drozdick      Added lportLastInvalidDLCI and
--                                 lportTrkProtState.
-- 01/11/95  Mike Drozdick      Added cardTrkProtState.
-- 01/16/95  Chris D'Souza      Added ds1 loopback objects
-- 01/19/95  Martin Jensen      Added version info objects to card group
-- 03/06/95  Chris D'Souza      Modified objects for short haul T1 cards
-- 03/10/95  John Moy           Added lportTrkTrafficMix.
-- 04/17/95  Jim Coronella      Added Atm pport and ckt objects.
-- 04/20/95  Jonathan West      fix diag code for hssi online diag tests
-- 04/28/95  Jim Martel         Added Atm ckt objects for cell rates
-- 05/05/95  John Moy           Added lport objects for routing QoS
--                                 changes
-- 05/25/95  Jonathan West      Added lport objects for QoS bandwidth
--                                 management
-- 05/31/95  Jonathan West      Added ckt objects for ATM traffic
-- 06/07/95  Ron Parker         Added:  cascsvc branch,
--                                 cascsvc.svcaddress group,
--                                 cascsvc.svcaddress.svcNodePrefixTable,
--                                 cascsvc.svcaddress.svcPrefixTable,
--                                 cascsvc.svcAddrTable,
--                                 cascsvc.svcaddress.svcDteUserPartTable,
--                                 cascsvc.svcmgt group,
--                                 cascsvc.svcmgt.svcFailedCallTable,
--                                 cascfr.ckt.cktLeafTable,
--                                 cascfr.lport.atmSigLportTable,
--                                 ATM lport entries,
--                                 svc ident stats upload lport entry,
--                                 svcFailedCall trap
-- 06/09/95  Phil Malloy        Added OAM objects to node, lport, and ckt
-- 06/09/95  Eric Li            Added Topaz IOM timing, various cell
--                                 counters, etc. in card and pport group.
-- 06/12/95  Ron Parker         Change indexing of SVC Failed Call Log to
--                                 port.inst.
--                              Change diag code object to octet string.
-- 06/19/95  Jonathan West      lport vars: lportInVAvailbw,
--                                 lportbwUNIPolicy.
-- 06/20/95  Jim Coronella      Added DS3 PLCP Enable/Disable.
-- 06/23/95  Ruediger Eckhard   Added B-STDX 9000 CBR/ATM-IWU objects.
-- 06/23/95  Eric Li            Added Topaz OC3 & STM1 IOPs.
-- 06/26/95  Stephen Bortolussi Added traps and lport objects for SMDS
--                                 Network trouble shooting.
-- 06/27/95  Mardy Marshall     Added:  Objects to the Node Group for
--                                 B-STDX 500 Application/Configuration
--                                 backup and restoration and management.
--                                 Objects to the Node Group and Traps for
--                                 B-STDX 500 System Timing Management.
-- 06/29/95  Oliver Lin         Added:  CardCpuFgUtil for Cpu foreground
--                                 utilization. And move the
--                                 cardTrkProtState.
-- 07/07/95  Ron Parker         Added svcRemotePrefixTable to
--                                 cascsvc.svcaddr.
-- 07/12/95  Mike Bernstein     Modified comment for pportDataRate
--                                 Added cktFailReason 29, 30
--                                 Updated lportErrType values through 19
-- 07/19/95  Ruediger Eckhard   Removed double defined
--                                 ckt{In,Out}CellsCLP[01]
--                                 Added pportOc3LoopConfig and
--                                 pportOc3LoopStatus
-- 07/25/95  Mike Drozdick      Added lportTrkProtFailureThreshold.
-- 07/31/95  Mike Drozdick      Added lportPtr.
-- 08/02/95  Eric Li            Added cardSystemPrimaryClockModeConfig &
--                                 cardSystemSecondaryClockModeConfig
-- 08/08/95  Ron Parker         Added svcConfigTable,moved
--                                 atmSigLportTable
--                                 to svcAtmConfigTable.
-- 08/14/95  Dan Bergman        Add cardISDNSigType, cardISDNChanId,
--                                 pportISDNIpBaseAddr,
--                                 lportISDNPoolUtil,
--                                 lportISDNIpAddrAssignFail&
--                                 the trap lportISDNIpAddrRej
--                                 Changed lportISDNRejCnt to
--                                 lportISDNIpAddr
-- 08/21/95  Dan Bergman        Moved cardISDNSigType and cardISDNChanId
--                                 up to where SWBuild
--                                 and SWBuildID were located
-- 08/31/95  RonParker          Expanded Q.93B and Q.SAAL statistics in
--                                 svcAtmConfigTable.
-- 09/13/95  Eric Li            Add traps for IOM timing section
-- 09/22/95  Mike Drozdick      Clarify definition of
--                                 lportLastInvalidDLCI.
-- 09/25/95  Larry Palmer       Removed svcaddr, netSvcAddrNetWidth,
--                                 netSvcAddrPortWidth,
--                                 netSvcAddrUserWidth,
--                                 svcAddressTable
-- 09/29/95  Eric Li            Add pportSonetSTM1Scramble & clarified
--                                 pportLaserStatus
-- 10/19/95  Al Plante          Add nodeTrapMaskMib and
--                                 nodeTrapMaskEnterprise
```

```
-- 10/23/95  Charles Zhou      Add cardNFBDEStatus in cardEntry
--
-- 10/31/95  Larry Palmer      Fixed miscellaneous typos and omissions
-- 11/03/95  Eric Li           Add pportEFCIMarking &
                                  pportAtmQOSTransmitMode
-- 11/20/95  Ashish Sagar      Changed lportXmitDiscardLow OBJECT-TYPE to
--                                lportAbsoluteThreshold OBJECT-TYPE
-- 11/21/95  Eric Li           Added ATM cell statistics for lport
-- 12/6/95   Al Plante         Add cardProductCode thru cardIOAMfgPN to
                                  card
-- 12/22/95  Phil Malloy       Added lport debug strings for ATM control
                                  ckts
-- 1/2/96    Al Plante         Update cardIOAType w/ presently defined
                                  types.
-- 01/18/95  Chris D'Souza     Changed object names in Cascade DS1 table.
-- 01/18/96  Ruediger Eckhard  Remove atmAAL1 from lportProtocol.
--                                It is not used for CBR cards.
-- 02/13/96  Ruediger Eckhard  Modify description of line build out of
--                                CBR T1 card from long haul to short
                                  haul
-- 02/21/96  Phil Malloy       Add card variables for internal
                                  diagnostics
-- 03/06/96  Sanjay Subbanna   Add traps for Billing CP+ Redundancy and
--                                Recovery
-- 03/13/96  Yui-Ting Lin      Add smds traffic [ga_mode, priority,
                                  color]to the net group
-- 03/26/96  James Ni          Added slide window rate enforcement and
                                  close loop congestion control related
                                  new variables in circuit entry (Delta
           -                      Bc, Delta-Be, Red frame
--                                percentage and slide window enable/
                                  disable).
-- 03/28/96  Ruediger Eckhard  Deprecate pportClockMasterChannel (pport
                                  75)
-- 04/01/96  Vijay Pandian     Added nodeLanIpMask, netRipStatus, and
--                                netRipSendHostRoutes.
-- 04/02/96  Ralph Atenasio    Added Authentication Domain table to
                                  'node' section and authentication
                                  params to 'lport'
                                  section. Added Authentication Fail
                                  trap.
                                  Added PPP MP/BACP/Echo-Request params
                                  to 'lport' section. Added MP and BACP
                                  traps.
-- 04/03/96  James Ni          added 'cktEntry 148'
-- 4/17/96   Ralph Atenasio    Changed IP address fail code/
                                  trap to more
--           Ken Chen general  'PPP Negotiation Fail' code.
--                                Eliminated Domain ID from Authen. Trap.
--                                Added ISDN-PPP traps.
-- 04/19/96  Stephen Bortolussi Added SMDS SSI logical
                                  port counter -
--                                lportSmdsCntDstGaSrcIsCascade.
-- 04/26/96 Ruediger Eckhard   Add lportCbrFifoHalfLength. Obsolete
--                                cktOspfCtd and cktOspfCdv for CBR.
-- 04/30/96  Mark Libby        Add cug group & tables
-- 04/30/96  Rumi S. Gonda     Add pportHECMode.
-- 05/01/96  Ron Parker        Add Address attribute objects
-- 05/01/96  Ashish Sagar      Modified description for
                                  lportTrkStaticDelay,
--                                and lportTrkDynamicDelay
-- 05/03/96  Paul Fay          Added chands1LoopChange trap
-- 05/03/96  Prasad Talwalkar  Added: Screen Table.
--                                New objects in Port table for
                                  screening.
-- 05/03/96  Marek Kotelba     Added new objects for NTM/NDC.
-- 05/06/96  Mark Libby        Add CUG status elements
-- 05/07/96  Bob Braisted      Add pportISDNChannelStatus element
-- 04/26/96  Ruediger Eckhard  Allow cktOspfCtd for CBR in 4.2
-- 05/15/96  Umesh Bhatt       Add two variables to manage SMDS-Ga
                                  Area Mask
-- 05/16/96  Paul Murray       Added pport objects to support Topaz T1
                                  ATM, and
--                                Cascade Performance Monitoring Tables
                                  for DS1
-- 05/22/96  James Ni          Added lportEntry 305 to 309.
-- 05/28/96  Ron Parker Deprecatelportatmconnectiontype
--                                Add lportConnectionType,
                                  lportAtmUniType
-- 06/08/96  Yui-Ting Lin      Add 8 SMDS counters to the lport table
--                                Num IA/GA Frames/Bytes in/out of ssi/
                                  dxi-sni
-- 06/10/96  Al Plante         Add 'sonetpm' group for ANSI T1.231
                                  support
-- 06/10/96  Paul Fay          Add FEAC codes to pportdsx3FEACStatus
-- 06/13/96  Ken Chen          Add new objects nodeAuthState,
--                                nodeAuthDomainID and
                                  nodeAuthFailReason,
--                                new trap nodeAuthenticationFailure.
-- 06/21/96  Michael Carr      added ATM billing objects
-- 06/24/96  Ruediger Eckhard  ATMU loopback not supported on ATM-IWU by
                                  HW
-- 06/25/96  Ken Chen          add nodeAuthLoginUser object
-- 06/05/96  Sushil Shelly     Add bici-11 to lportAtmProtocol
--                                Add lportAtmCellType - oid 85
-- 06/26/96  James Ni          Changed the comment of lportEntry 305.
-- 07/11/96  Paul Fay          Add chands1AlarmStateChange trap
-- 07/12/96  Marek Kotelba     Added NRTS processor support.
-- 07/15/96  Ken Chen          Modify trap 43 lportISDNCallRej
-- 07/17/96  James Ni          Added LportEntry 318.
-- 07/17/96  Ken Chen          Changed variables of trap 102
-- 07/25/96  Mike Drozdick     Added Multilink PPP cktEntry objects.
-- 08/01/96  James Ni          Revised the descriptions of lportEntry
                                  307 and
--                                cktEntry 140 to 143
-- 08/14/96  Prasasth Palnati  Added objects for N+1 Power Supply system
--                                Node Group: 115 to 118
--                                Traps group: 113 and 114
```

```
-- 08/15/96  Mike Sheehan      Added to PPort Bchannel timer value for
                                  ISDN.
-- 08/22/96  Rich Leary        Added for Trap bindings: node 119 120
-- 09/03/96  Ken Chen          Added lportPrivateNet and lportCustomerID
--                                to Trap 19 and 30, cktPrivateNet and
--                                cktCustomerID to Trap 20
-- 09/03/96                    James Ni Revised some congestion control
                                  related entries.
-- 09/08/96                    Rich Leary Added nodeTrapTxRate,
                                  nodeTrapMaskSeverity,
--                                nodeAlarmRelayStatus,
                                  nodeTrapsDiscardedMajor,
--                                nodeTrapsDiscardedMinor,
--                                nodeTrapDiscardedNonalarm.
--                                Added nodeTrapSeverity &
                                  nodeTrapSequenceNumber
--                                to all trap bindings
-- 09/12/96  Nick Elkherj      Added all the tables related
                                  to the SDLC FRAD
--                                project.  They are, snatlAdminTable,
--                                snasdlcPortTable, snasdlcLsTable,
--                                snallcPortAdminTable,
                                  snallcPortOperTable,
--                                snallcPortStatsTable,
                                  snallcLsAdminTable,
--                                snallcLsOperTable, snallcLsStatsTable.
-- 09/17/96  Rich Leary        Added trap masks to nodeNMS table.
--                                Changed nodeTrapSequenceNumber to
                                  Counter
-- 09/17/96  Dan Bechar        Added branch point for MPT.  (Rest of
                                  branch
--                                is in mib/cascmpt.mib.)
-- 10/09/96  Gabe Montero      snallcPortOperStatus enums incorrect
-- 10/16/96  Mark Libby        Add in atmckt group
--                                Add in ctlCktTable
-- 10/17/96  James Ni          revised the cktEntry 146 and 147
-- 10/23/96  James Ni          revised the comments of some cktEntry
-- 10/23/96  Gabe Montero      Add new entries to snatlAdminEntry,
--                                snatlOperEntry, and snallcLsAdminEntry
-- 10/24/96  James Ni          revised lportEntry 79, 80, 81.
-- 11/08/96  Uwe Lepa          add new entries pportAAL5CRC32Error,
--                                pportAAL5CPIError,
                                  pportAAL5LengthError,
--                                pportAAL5ReassemblyTimerError,
--                                pportAAL5MaxNrSegError
-- 11/11/96  Yui-Ting Lin      fix lportSmdsIaScrnMap comment
-- 11/15/96  Gabe Montero      change snatlAdminIdBlkNum from integer to
--                                octet string
-- 12/18/96  Ralph Atenasio    Adding IP QoS Option (enable/disable) to
                                  lport
-- 12/19/96  Greg Jigarjian    Adding NTP object to the Node group
--                                and a NTP table to the card group
-- 12/30/96  Ralph Atenasio    Removing IP QoS option (enable/disab) fm
                                  lport
-- 01/09/97  Jecko Li          Added FaultServer Alarm Trap Mib
-- 01/10/97  James Ni          Added some lport and ckt entries for 9000
                                  integrated
--                                QoS development.
-- 01/29/97                    James Ni Deleted the ServiceClassType
                                  entry from ckt entry
--                                table. Revised cktATMQoS, cktATMRQoS,
                                  cktPriority
--                                and cktRPriority entries.
-- 01/30/97  Jeff Rosenberg    Added farend _loopback_initiated to
                                  loopback status.
-- 01/31/97  Doug Grote        Added nodeLanIdleTimeout.
-- 02/03/97  James Ni          Added cktFrameSize and cktRFrameSize
                                  entries.
-- 02/07/97  James Ni          Revised the descriptions of cktDeltaBc
--                                cktDeltaBe etc.
-- 02/07/97  Sreeni. G         Changed lnkTrkDynamicDelay to read-only.
-- 02/07/97  Arvind P.         Addition of SSI Feeder address definition.
-- 02/12/97  James Ni          fix the description display bug for
                                  cktRedFrPcnt
--                                and cktRedFrRPct.
-- 02/13/97  James Ni          Added CBR rt_VBR nrt_VBR and UBR
                                  specifying entries in
--                                lport table entry when the lport is
                                  specified as
--                                multiple-service mode.
-- 02/14/97  James Ni          reveised lportCongestionCheckInterval and
--                                lportCongestionClearDelay.
-- 02/19/97  Yui-Ting Lin      Update cardIOAType w/ BSTDX UIO-V35 and
                                  UIO-X21
-- 02/20/97  P. Fay            Add chassis clock source to
                                  pportXmitClock/pportRecvClock.
--                                Add unstructured to pportDs1LineType.
-- 02/24/97  Paul Fay          Add E1 75/120 ohm IOA types to
                                  cardIOAType for BSTDX
-- 02/26/97  Paul Fay          Add to e1 external clock types to
                                  cardTransmitClockConfig

*****************************************************************************
```

CASCADE-MIB DEFINITIONS

IMPORTS

```
        enterprises, IpAddress, Counter,
        Gauge, TimeTicks
                FROM RFC1155-SMI
        DisplayString, ifInErrors
                FROM RFC1213-MIB
        OBJECT-TYPE
                FROM RFC-1212
        TRAP-TYPE
                FROM RFC-1215;

--  This MIB module uses the extended OBJECT-TYPE macro as
--  defined in RFC 1212
```

OID Registration Pointers

```
cascade         OBJECT IDENTIFIER ::= { enterprises 277 }
cascfr          OBJECT IDENTIFIER ::= { cascade 1 }
cascsmds        OBJECT IDENTIFIER ::= { cascade 2 }
namebinding     OBJECT IDENTIFIER ::= { cascade 3 }
isdnaddr        OBJECT IDENTIFIER ::= { cascade 4 }
cascsvc         OBJECT IDENTIFIER ::= { cascade 5 }
software        OBJECT IDENTIFIER ::= { cascade 6 }
provserver      OBJECT IDENTIFIER ::= { cascade 9 }

net             OBJECT IDENTIFIER ::= { cascfr 1 }
ase             OBJECT IDENTIFIER ::= { cascfr 2 }
node            OBJECT IDENTIFIER ::= { cascfr 3 }
pport           OBJECT IDENTIFIER ::= { cascfr 4 }
lport           OBJECT IDENTIFIER ::= { cascfr 5 }
ckt             OBJECT IDENTIFIER ::= { cascfr 6 }
card            OBJECT IDENTIFIER ::= { cascfr 7 }
ds1             OBJECT IDENTIFIER ::= { cascfr 8 }
svcaddr         OBJECT IDENTIFIER ::= { cascfr 9 }
smdsaddr        OBJECT IDENTIFIER ::= { cascsmds 1 }
svcaddress      OBJECT IDENTIFIER ::= { cascsvc 1 }
svcmgt          OBJECT IDENTIFIER ::= { cascsvc 2 }
```

Textual Conventions

```
Index ::= INTEGER--  1..ifNumber
```

The Network Group

```
--  The variables that are relevant to a Cascade network
netMask OBJECT-TYPE
        SYNTAX IpAddress
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The IP subnet mask for the network.  The default is
                255.255.0.0 (class B) which allows 400 nodes."
        ::= { net 1 }

netNumber OBJECT-TYPE
        SYNTAX IpAddress
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The IP network number for the network."
        ::= { net 2 }

netDlciAddrSig OBJECT-TYPE
        SYNTAX INTEGER {
        globalAddr (1),
        localAddr (2)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This variable states the DLCI addressing significance."
        ::= { net 3 }

netMaxSegsize OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This variable states the maximum segment size for the
                network."
        ::= { net 4 }

netSmdsAreaMaskStart OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "This variable states the starting digit number for
                defining the start of the Area Mask for SMDS Addresses."
        ::= { net 9 }
```

```
netSmdsAreaMaskDigits OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "This variable states the number of digits used for
                defining the Area Mask for SMDS Addresses."
        ::= { net 10 }

netCaCType OBJECT-TYPE
        SYNTAX INTEGER {
                cacCASCADE        (1),
                cacCUSTOMIZED     (2),
                cacCUSTOMIZED_NRT (3)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                " The CAC (Connection Admission Control) algorithm can be
                set to any of 3 algorithms:
                    1. The Cascade default algorithm.
                    2. The custom CAC algorithm where the bandwidth of
                    both VBRrt and VBRnrt circuits is customized.
                    3. The custom_nrt CAC algorithm where only the
                    bandwidth of the VBRnrt circuits is customized; the
                    cascade default algorithm is
                    applied for the bandwidth computation of the VBRrt
                    circuits."
                ::= { net 11 }

netCaCCLRobjectiveQoS2 OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The Cell loss ratio objective for QoSclass2 (VBRrt) used
                by the Cascade default CAC."
        ::= { net 12 }

netCaCCLRobjectiveQoS3 OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The Cell loss ratio objective for QoSclass3 (VBRnrt) used
                by the Cascade default CAC."
        ::= { net 13 }

netCaCPortScaleFactorDS3 OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "DS3 port Scale factor used by the customized CAC. The
                actual scale factor used is this value divided by 100 to
                account for lack of float definitions in MIBs"
         ::= { net 14 }

netCaCPortScaleFactorOC3 OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "OC3 port Scale factor used by the customized CAC. The
                actual scale factor used is this value divided by 100 to
                account for lack of float definitions in MIBs"
         ::= { net 15 }

netCaCScrCustomTable OBJECT-TYPE
         SYNTAX SEQUENCE OF NetCaCScrCustomEntry
         ACCESS not-accessible
         STATUS mandatory
         DESCRIPTION
                "A table of SCR upper limits, corresponding scale factors
                and maximum supported MBS's used by the Customized CAC."
        ::= { net 16 }

netCaCScrCustomEntry OBJECT-TYPE
        SYNTAX  NetCaCScrCustomEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                " The CaC SCR customization table entry "
        INDEX   { netCaCScrUpperLimit }
        ::= { netCaCScrCustomTable 1 }

NetCaCScrCustomEntry ::=
        SEQUENCE {
                netCaCScrUpperLimit  INTEGER,
                netCaCScrScaleFactor INTEGER,
                netCaCEntryStatus    INTEGER,
        }

netCaCScrUpperLimit OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                " SCR upper limit value. "
        ::= { netCaCScrCustomEntry 1 }
```

```
netCaCScrScaleFactor OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The SCR Scale factor used by the customized CAC. The
                actual scale factor used is formed by dividing over 100 to
                account for lack of float definitions in MIBs"
        ::= { netCaCScrCustomEntry 2 }

netCaCEntryStatus OBJECT-TYPE
        SYNTAX INTEGER {
                configured (1), -- this entry has been configured by NMS
                invalid (2)    -- this entry shall be deleted
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The status of this entry."
        ::= { netCaCScrCustomEntry 3 }

netCaCMaxMBS OBJECT-TYPE
         SYNTAX  INTEGER
         ACCESS  read-write
         STATUS  mandatory
         DESCRIPTION
                " The Maximum MBS supported under the customized CAC. A
                maximum MBS value is configured for each range of SCR
                values. When a VC is being set-up, this value is compared
                to the VC's MBS.If the VC's MBS is higher, the circuit is
                rejected."
        ::= { netCaCScrCustomEntry 4 }

netResetCaCTable OBJECT-TYPE
        SYNTAX  INTEGER {
                     reset-cac-tbl (1)
                    }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Setting to 1 causes a reset of the customized cac Table
                in the PRAM."
        ::= { net 17 }

netSmdsTrafficMode OBJECT-TYPE
        SYNTAX  INTEGER {
                bellcore (1),
                cascade (2)
        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "This variable indicates the ga mode for SMDS traffic."
        ::= { net 18 }

netSmdsTrafficPriority OBJECT-TYPE
        SYNTAX  INTEGER {
                high (1),
                medium (2),
                low (3)
        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "This variable indicates the priority of SMDS traffic."
        ::= { net 19 }

netSmdsTrafficColor OBJECT-TYPE
        SYNTAX  INTEGER {
                green (1),
                amber (2),
                red (3)
        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "This variable indicates the color for SMDS traffic."
        ::= { net 20 }

netCaCPortScaleFactorE3 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "E3 port Scale factor used by the customized CAC. The
                actual scale factor used is this value divided by 100 to
                account for lack of float definitions in MIBs"
        ::= { net 21 }
```

```
netRipStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                off (0),
                on (1)
        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The configured state of RIP.
                 The default state is off."
        ::= { net 22 }

netRipSendHostRoutes OBJECT-TYPE
        SYNTAX  INTEGER {
                off (0),
                on (1)
        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The configured state of SendHostRoutes.
                 The default state is off."
        ::= { net 23 }

netCaCPortScaleFactorOC12 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "OC12 port Scale factor used by the customized CAC. The
                actual scale factor used is this value divided by 100 to
                account         for lack of float definitions in MIBs"
        ::= { net 24 }

netCaCPortScaleFactorT1 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "T1 port Scale factor used by the customized CAC. The
                actual scale factor used is this value divided by 100 to
                account for lack of float definitions in MIBs"
        ::= { net 25 }

netSmdsGaAreaMaskStart OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "This variable defines the starting digit number of the
                SMDS Group Address where the Ga Area Mask begins."
  ::= { net 26 }

netSmdsGaAreaMaskDigits OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "This variable states the number of digits used in SMDS
                Group Address Area Mask."
  ::= { net 27 }

netCaCCDVobjectiveQoS1 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The Cell delay variation objective (in microseconds) for
                 QoSclass1 (CBR) used by the Cascade default CAC."
        ::= { net 28}

netCaCCDVobjectiveQoS2 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The Cell delay variation objective (in microseconds) for
                 QoSclass2 (VBRrt) used by the Cascade default CAC."
        ::= { net 29}

 netCaCCDVCellFractionQoS1 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The fraction of QoSClass1 (CBR) cells that can exceed the
                CDV objective  used by the Cascade default CAC. A value
                of x indicates that only 1E-x of the Cells can exceed the
                specified CDV objective"
        ::= { net 30}

netCaCCDVCellFractionQoS2 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The fraction of QoSClass2 (VBRrt) cells that can exceed
                the CDV objective  used by the Cascade default CAC. A
                value of x indicates that only 1E-x of the Cells can
                exceed the specified CDV objective"
        ::= { net 31}
```

OSPF ASE Device and Host Table

```
--  For NMS paths (Currently an External Device is always accessible via
--  Ethernet, whereas a Host is always accessible via SLIP.)

aseTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF AseEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A list of ASE entries."
        ::= { ase 1 }

aseEntry OBJECT-TYPE
        SYNTAX  AseEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "An ASE entry contains an external NMS path accessible from
                the node."
        INDEX   { aseAddr }
        ::= { aseTable 1 }

AseEntry ::=
        SEQUENCE {
                aseAddr
                    IpAddress,
                aseMask
                    IpAddress,
                aseDefaultGwAddr
                    IpAddress,
                aseMetricType
                    INTEGER,
                aseAdminStatus
                    INTEGER,
                aseIfIndex
                    Index,
                aseDlci
                    INTEGER
        }

aseAddr OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "IP Address for an external NMS."
        ::= { aseEntry 1 }

aseMask OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "IP Address mask for an external NMS.  The default value
                is 255.255.255.255."
        ::= { aseEntry 2 }

aseDefaultGwAddr OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Default Gateway IP Address for reaching the external
                NMS.  This field is not applicable to NMSs which are
                reachable via SLIP or on the same Ethernet."
        ::= { aseEntry 3 }

aseMetricType OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Type of metric (1 or 2) for each external NMS.
                The default is 1.  (This field is not applicable to SLIP-
                based hosts.)"
        ::= { aseEntry 4 }
aseAdminStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                    invalid (0),
                    valid (1)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The desired state of the entry."
        ::= { aseEntry 5 }

aseIfIndex OBJECT-TYPE
        SYNTAX  Index
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The interface value of the corresponding MIB-II ifEntry."
        ::= { aseEntry 6 }
aseDlciOBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The management dlci through which the NMS communicates
                 to the switch."
        ::= { aseEntry 7 }
```

The Node Group

```
--  The variables that configure a node

nodeIpAddr OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The inband IP address of the node in the Cascade internal
                trunk IP network."
        ::= { node 1 }

nodeLanIpAddr OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The inband (Ethernet) IP address of the node for a NMS
                to access the node and the internal trunk IP network."
        ::= { node 2 }

nodeNMSTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF NodeNMSEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A list of NMS entries."
        ::= { node 3 }

nodeNMSEntry OBJECT-TYPE
        SYNTAX  NodeNMSEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A NMS entry contains an IP address of a NMS which has
                extended privilages."
        INDEX   { nodeNMSIndex }
        ::= { nodeNMSTable 1 }

NodeNMSEntry ::=
        SEQUENCE {
                nodeNMSIndex
                    Index,
                nodeNMSIpAddr
                    IpAddress,
                nodeNMSflags
                    INTEGER,
                nodeNMSTrapMaskMIBII
                    OCTET STRING,
                nodeNMSTrapMaskSeverity
                    OCTET STRING,
                nodeNMSTrapMaskEnterprise
                    OCTET STRING,
                nodeNMSTrapSequence
                    Counter

                }

nodeNMSIndex OBJECT-TYPE
        SYNTAX  Index
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The IP address entry of a primary Network Management
                Station."
        ::= { nodeNMSEntry 1 }
```

```
nodeNMSIpAddr OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The IP address of the primary Network Management Station.
                It is to this address that Traps and TFTP requests
                will be directed."
        ::= { nodeNMSEntry 2 }

nodeNMSflags OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Community member flag, 0x1 = read/write, 0x2 = trap
                enable"
        ::= { nodeNMSEntry 3 }

nodeNMSTrapMaskMIBII OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Trap Masks for MIB II defined traps"
        ::= { nodeNMSEntry 4 }

nodeNMSTrapMaskSeverity OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Trap Masks for trap severity levels"
        ::= { nodeNMSEntry 5 }

nodeNMSTrapMaskEnterprise OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Trap Masks for Cascade enterprise traps"
        ::= { nodeNMSEntry 6 }

nodeNMSTrapSequence OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Trap Sequence number for retransmission"
        ::= { nodeNMSEntry 7 }
```

```
nodeState OBJECT-TYPE
        SYNTAX  INTEGER {
                    down (1),
                    initializing (2),
                    active (3),
                    marginal (4),
                    testing (5)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The state of the switch. The switch needs to await the
                state to be set to active before being fully operational
                after cold boot, when the NMS is reachable."
        ::= { node 4 }

nodePollStatus OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The consolidated node status in response to keep-alive
                polls.  It is an octet string of a proprietary format"
        ::= { node 5 }

nodeModel OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The part number of the node."
        ::= { node 6 }

nodeSerial OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The serial number (i.e., Ethernet Addr) of the switch."
        ::= { node 7 }

nodeSwRev OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The software revision number (major.minor)."
        ::= { node 8 }
```

```
nodeHwRev OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The hardware revision number (major.minor)."
        ::= { node 9 }

nodeEpromRev OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The EPROM firmware revision number (major.minor)."
        ::= { node 10 }

nodeCpuUtil OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The CPU utilization percentage for packet processing."
        ::= { node 11 }

nodePsAStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                    up (1),
                    down (2),
                    marginal (4)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The status of the power supply A.  For a detailed
                explanation for the power supply being either in the down
                or marginal state, refer to the nodePsADiagCode object."
        ::= { node 12 }

nodePsBStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                    up (1),
                    down (2),
                    marginal (4)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The status of the power supply B.  For a detailed
                explanation for the power supply being either in the down
                or marginal state, refer to the nodePsBDiagCode object."
        ::= { node 13 }

nodeFanTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF NodeFanEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A list of fan entries."
        ::= { node 14 }

nodeFanEntry OBJECT-TYPE
        SYNTAX  NodeFanEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A Fan entry contains the status of a fan."
        INDEX   { nodeFanIndex }
        ::= { nodeFanTable 1 }

NodeFanEntry ::=
        SEQUENCE {
                nodeFanIndex
                    INTEGER,
                nodeFanStatus
                    INTEGER,
                nodeFanSpeed
                    Gauge
        }

nodeFanIndex OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A unique value for each fan."
        ::= { nodeFanEntry 1 }

nodeFanStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                    up (1),
                    down (2),
                    marginal (3)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The status of the fan."
        ::= { nodeFanEntry 2 }

nodeFanSpeed OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The speed in Revolutions Per. Minute (RPM) of the fan."
        ::= { nodeFanEntry 3 }
```

```
nodeMemoryUtil OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The percentage of system memory utilization on this
                intelligent card."
        ::= { node 15 }

nodeMemoryUsage OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The system memory utilization, in terms of free bytes, for
                this intelligent card."
        ::= { node 16 }

nodeMaxFramesize OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The maximum frame size supported by the node.
                The default is 4096."
        ::= { node 17 }

nodeQOSPollTimer OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The timer (seconds) for generating QOS polling packets."
        ::= { node 18 }

nodeActivePvcs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of currently active PVCs on the node."
        ::= { node 19 }

nodeInactivePvcs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of currently inactive PVCs on the node."
        ::= { node 20 }

nodePendingPvcs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of currently call pending PVCs on the node."
        ::= { node 21 }

nodeInOctets OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of octets received on the node, including
                framing bytes."
        ::= { node 22 }

nodeInPkts OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of packets received on the node."
        ::= { node 23 }

nodeOutOctets OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of octets transmitted out of the node,
                including framing bytes."
        ::= { node 24 }

nodeOutPkts OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of packets requested to be transmitted
                out of the node, including those that were discarded or
                not sent."
        ::= { node 25 }

nodeSwFilename OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Name of the file to be downloaded."
        ::= { node 26 }
```

```
nodeRebootAfterLoad OBJECT-TYPE
        SYNTAX  INTEGER {
                    no-reboot (0),
                    reboot (1)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Instructs the device to either automatically reboot after
                the successful scheduled download or not."
        ::= { node 27 }

nodeSwToLoad OBJECT-TYPE
        SYNTAX  TimeTicks
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Instructs the device to initiate a software download
                (via tftp) request to the master management station after
                the specified time ticks have elapsed."
        ::= { node 28 }

nodeSwLoadState OBJECT-TYPE
        SYNTAX  INTEGER {
                    inactive (1),
                    pending (2),
                    active (3),
                    failed (4)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The state of the software download request.  A download
                which is in either the PENDING, ACTIVE or FAILED state
                can be terminated by setting this object to INACTIVE."
        ::= { node 29 }

nodePrFilename OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Name of the PRAM image file to be downloaded."
        ::= { node 30 }

nodePrToLoad OBJECT-TYPE
        SYNTAX  TimeTicks
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Instructs the device to initiate a PRAM image file
                download (via tftp) request to the master management
                station after  the specified time ticks have elapsed."
        ::= { node 31 }

nodePrLoadState OBJECT-TYPE
        SYNTAX  INTEGER {
                    inactive (1),
                    pending (2),
                    active (3),
                    failed (4)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The state of the PRAM download request.  A download
                which is in either the PENDING, ACTIVE or FAILED state
                can be terminated by setting this object to INACTIVE."
        ::= { node 32 }

nodeToWarmboot OBJECT-TYPE
        SYNTAX  TimeTicks
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Instructs the switch to initiate a system warm boot after
                the specified time ticks have elapsed. A value of 0
                indicates cancellation of the previously scheduled re-boot
                request."
        ::= { node 33 }

nodeToColdboot OBJECT-TYPE
        SYNTAX  TimeTicks
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Instructs the switch to initiate a system cold boot after
                the specified time ticks have elapsed. A value of 0
                indicates cancellation of the previously scheduled re-boot
                request."
        ::= { node 34 }

nodeToRedundant OBJECT-TYPE
        SYNTAX  TimeTicks
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "If redundancy is configured, instructs the node acting as
                Backup to disable the Active node and take over as the
                Active node. The disabling is initiated after the
                specified time ticks have elapsed. A value of 0 indicates
                cancellation of the previously scheduled request."
        ::= { node 35 }
```

```
nodeInitiateBulkStats OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Initiates a Bulk Statistics poll and via TFTP, transfer
                the data to the specified IP address."
        ::= { node 36 }

nodeDiagNonFatalSource OBJECT-TYPE
        SYNTAX  INTEGER {
                power-on-diagnostics (1),
                background-diagnostics (2),
                fault (3),
                frame-heap (4),
                redundancy (5),
                system-level (6),
                card-level (7),
                i960-data-structures (8),
                general (9),
                data-alignment (10),
                device-driver-level (11)
        }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Source who reported last non-fatal error."
        ::= { node 37 }

nodeDiagNonFatalTime OBJECT-TYPE
        SYNTAX  TimeTicks
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Time the last non-fatal error was reported."
        ::= { node 38 }

nodeDiagNonFatalErrMajor OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Major error code of last non-fatal error."
        ::= { node 39 }

nodeDiagNonFatalErrMinor OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Minor error code of last non-fatal error."
        ::= { node 40 }nodeDiagNonFatalStr OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Ascii string describing last non-fatal error."
        ::= { node 41 }

nodeDiagFatalSource OBJECT-TYPE
        SYNTAX  INTEGER {
                power-on-diagnostics (1),
                background-diagnostics (2),
                fault (3),
                frame-heap (4),
                redundancy (5),
                system-level (6),
                card-level (7),
                i960-data-structures (8),
                general (9),
                data-alignment (10),
                device-driver-level (11)
        }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Source who reported last fatal error."
        ::= { node 42 }

nodeDiagFatalTime OBJECT-TYPE
        SYNTAX  TimeTicks
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Time the last fatal error was reported."
        ::= { node 43 }

nodeDiagFatalErrMajor OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Major error code of last fatal error."
        ::= { node 44 }
```

```
nodeDiagFatalErrMinor OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Minor error code of last fatal error."
        ::= { node 45 }
nodeDiagFatalStr OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Ascii string describing last fatal error."
        ::= { node 46 }

nodeDiagFatalReboots OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Number of times the switch has re-booted since last
                fatal error was reported."
        ::= { node 47 }

nodeDiagFatalAddress OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Address switch was executing when it encountered fatal
                error."
        ::= { node 48 }

nodeDiagBackgroundPasses OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Number of passes made by the background diagnostics."
        ::= { node 49 }

nodeDiagBackgroundFailures OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Number of failures discovered by background diagnostics."
        ::= { node 50 }

nodeDiagBackgroundSuccesses OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Number of successful passes by background diagnostics."
        ::= { node 51 }
nodeDiagLEDReset OBJECT-TYPE
        SYNTAX  INTEGER {
                state-unchanged (0),
                state-to-active (1)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Change the node state from marginal to active.  Change the
                LED from yellow to green."
        ::= { node 52 }

nodeDiagPowerExtensive OBJECT-TYPE
        SYNTAX  INTEGER {
                fast-tests (0),
                extensive-tests (1)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Should power on diagnostics do slower, extensive testing,
                or fast testing?"
        ::= { node 53 }

nodePortPoll OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The consolidated port status in response to keep-alive
                polls."
        ::= { node 54 }

nodeMaxTelnetConsole OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The number of concurrent telnet consoles allowed. The
                default is 2. The maximum is 5. Set it 0 to disable telnet
                access."
        ::= { node 55 }
```

```
nodeConsoleTimeout OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Idle-logoff timer in minutes. The default is 5 minutes,
                and the maximum can be set is 60 minutes. To disable
                idle-logout, set this to 0."
        ::= { node 56 }

nodeConsoleTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF NodeConsoleEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A list of active consoles on the node, logged in through
                telnet or serial port."
        ::= { node 57 }

nodeConsoleEntry OBJECT-TYPE
        SYNTAX  NodeConsoleEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "Information of a console."
        INDEX   { nodeConsoleIndex }
        ::= { nodeConsoleTable 1 }

NodeConsoleEntry ::=
        SEQUENCE {
                nodeConsoleIndex
                    INTEGER,
                nodeUserName
                    OCTET STRING,
                nodeUserFrom
                    IpAddress,
                nodeConsoleAccessMode
                    INTEGER,
                nodeConsoleUptime
                    TimeTicks
        }

nodeConsoleIndex OBJECT-TYPE
        SYNTAX  INTEGER (1..6)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A positive integer to identify a console"
        ::= { nodeConsoleEntry 1 }

nodeUserName OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The login user name up to 31 characters. There is no
                 validation of the name, i.e., any string is acceptable."
        ::= { nodeConsoleEntry 2 }

nodeUserFrom OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The IP address from which the user logged in. 0.0.0.0 if
                the user logged in through serial port."
        ::= { nodeConsoleEntry 3 }

nodeConsoleAccessMode OBJECT-TYPE
        SYNTAX  INTEGER {
                readOnly (1),
                readWrite (2)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The access mode of the console. By default, a user is in
                READ ONLY mode."
        ::= { nodeConsoleEntry 4 }

nodeConsoleUptime OBJECT-TYPE
        SYNTAX  TimeTicks
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "How long the console has been active."
        ::= { nodeConsoleEntry 5 }

nodePsADiagCode OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The failure code of the power supply A.  A non-zero value
                indicates one or more failures.  Refer to the hardware
                manual for a description of the failure code."
        ::= { node 58 }
```

```
nodePsBDiagCode OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The failure code of the power supply B.  A non-zero value
                 indicates one or more failures.  Refer to the hardware
                 manual for a description of the failure code."
        ::= { node 59 }nodeFrameMemoryUtil OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The percentage of frame memory utilization on this
                intelligent card."
        ::= { node 60 }

nodeFrameMemoryUsage OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The frame memory utilization, in terms of free bytes, for
                this intelligent card."
        ::= { node 61 }

nodeCapability OBJECT-TYPE
        SYNTAX  INTEGER {
                    frame-relay (1),
                    smds (2)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The capability of this node."
        ::= { node 62 }

nodeSvcLastCallFailure OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A textual string describing the last call failure."
        ::= { node 63 }

nodeRerouteDelay OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The number of seconds delay between each reroute batch
                event."
        ::= { node 64 }
nodeRerouteCount OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The number of virtual circuits to request a reroute for
                duing a single reroute batch event."
        ::= { node 65 }
```

```
nodeFileTransferRequest OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "This object is used to initiate file transfers to and from
                 the switch. A management station can initiate a file
                transfer by settting this object of text string in the
                following format:

                      operation filename time-to-wait target-host protocol

                where
                    operation   -[get | put] (from switch's
                                 perspective),
                    filename    -name.ext
                where
                    name        -full path excluding file
                                 extension
                    ext         -file extension formatted as
                                 follows
                    time-to-wait-a number in seconds
                    target-host-the IP address of target host
                    protocol    -[tftp | ftp]

                 File extensions may take the following form:
                    Cxx or cxx  - Configuration image for card xx
                    Dxx or dxx  - Memory dump for card xx
                    Syyyy or syyyy- SVC identification stats for port yyyy
                    Fyyyy or fyyyy- SVC failure log for port yyyy
                    Iyyyy or iyyyy- ILMI dynamic addresses for port yyyy

                 Binary file formats are available upon request.

                Multiple transfers can be specified by repeating above
                with between each transfer specifier.

                 Examples:

                   o  put configuration.10 5 152.148.10.100 tftp

                      will start a transfer of configuration information
                      from card 10 to host 152.148.10.100 in 5 seconds.

                To cancel the transfer request, set this object to a null
                string."
        ::= { node 66 }

nodeFileTransferStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                complete (1),  -- transfer complete or no outstanding
                               -- transfer request
                active (2),    -- transfer is in progress
                failed (3),    -- transfer failed - generic
                                  error
                invalid (4),   -- invalid request
                filename(5),   -- file name error or file non-
                                  existent
                timeout (6),   -- transfer timed out
                canceled (7),  -- transfer is cancelled
                file-not-available (8) -- requested file is not
                                          available
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The status of the file tarnsfer."
        ::= { node 67 }

nodeTime OBJECT-TYPE
        SYNTAX  TimeTicks
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The current wall-clock time that is set on the switch in
                Universal Coordinated Time (UCT).  The value is the number
                of seconds since 00:00:00 UCT January 1, 1970."
        ::= { node 68 }

nodeBillingAPAddress OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The IP Address of the Adjunct Billing Processor that is
                servicing this switch."
        ::= {node 69 }

nodeBillingAPUsername OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The Username of the account on the Adjunct Billing
                Processor to which usage data will be transferred via FTP."
        ::= { node 70 }
```

```
nodeBillingAPPassword OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The Password corresponding to the account identified by
                nodeBillingAPUsername.  A NULL string is returned when
                read."
        ::= { node 71 }

nodeBillingSwAPCommsFailures OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The number of times that communication from the switch to
                the Adjunct Billing Processor has failed during the current
                aggregation period.  A failure signifies failure of a file
                transfer operation to the Adjunct Processor."
        ::= { node 72 }

nodeBillingTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF NodeBillingEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A table of Billing System manageable objects, indexed by
                service."
        ::= { node 73 }

nodeBillingEntry OBJECT-TYPE
        SYNTAX  NodeBillingEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A nodeBillingEntry contains a set of statistics that
                summarize the Billing System performance for a particular
                service."
        INDEX { nodeBillingService }
        ::= {nodeBillingTable 1 }

NodeBillingEntry ::=
        SEQUENCE {
                nodeBillingService
                    INTEGER,
                nodeBilling
                    INTEGER,
                nodeBillingAggrPeriod
                    TimeTicks,
                nodeBillingCurAggrPeriodStart
                    TimeTicks,
                nodeBillingCurAggrPeriodEnd
                    TimeTicks,
                nodeBillingCollection
                    TimeTicks,
                nodeBillingDailyProcessing
                    INTEGER,
                nodeBillingDPTime
                    TimeTicks,
                nodeBillingUsageRecOvflWarnings
                    Counter,
                nodeBillingTotalUsageRecOvflWarnings
                    Counter,
                nodeBillingBillableUsageEvents
                    Counter,
                nodeBillingNonBillableUsageEvents
                    Counter,
                nodeBillingUsageRecCreated
                    Counter,
                nodeBillingTotalUsageRecCreated
                    Counter,
                nodeBillingUsageRecCrFailures
                    Counter,
                nodeBillingTotalUsageRecCrFailures
                    Counter,
                nodeBillingUsageRecSent
                    Counter,
                nodeBillingTotalUsageRecSent
                    Counter,
                nodeBillingUsageDataStoreFull
                    Counter,
                nodeBillingTotalUsageDataStoreFull
                    Counter,
                nodeBillingAdminAction
                    INTEGER
        }

nodeBillingService OBJECT-TYPE
        SYNTAX  INTEGER {
                    smds (1)
                    }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The service being reported.  This is the index into the
                table.  Additional table rows for Frame Relay and ATM may
                be provided in the future."
        ::= {nodeBillingEntry 1 }
```

```
nodeBilling OBJECT-TYPE
        SYNTAX  INTEGER {
                    disabled (1),
                    enabled  (2)
                    }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Switch to enable and disable billing on the STDX.

                When the value of nodeBilling is 'enabled', the value of a
                logical port's lportBilling object will take precedence.
                When the value of nodeBilling is 'disabled', it overrides
                all logical ports' lportBilling objects and billing is
                disabled across the entire switch.

                The default value of this object is 'disabled'."
        ::= {nodeBillingEntry 2 }

nodeBillingAggrPeriod OBJECT-TYPE
        SYNTAX  TimeTicks
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Defines the length of the Billing Aggregation Period in
                 seconds.  The default value is 15 minutes.  The minimum
                value is 10 minutes."
        ::= {nodeBillingEntry 3 }

nodeBillingCurAggrPeriodStart OBJECT-TYPE
        SYNTAX  TimeTicks
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The start-time of the current aggregation period, given by
                the number of seconds since 00:00:00 UCT January 1, 1970."
        ::= {nodeBillingEntry 4 }

nodeBillingCurAggrPeriodEnd OBJECT-TYPE
        SYNTAX  TimeTicks
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The end-time of the current aggregation period, given by
                the number of seconds since 00:00:00 UCT January 1, 1970."
        ::= {nodeBillingEntry 5 }

nodeBillingCollection OBJECT-TYPE
        SYNTAX  TimeTicks
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Defines how often the Data Aggregation process on the CP
                is to poll the IOPs for usage data.  The default value is
                30 seconds.  The range is unbounded, but practical use will
                dictate a range from 30 seconds to 1 minute."
        ::= {nodeBillingEntry 6 }

nodeBillingDailyProcessing OBJECT-TYPE
        SYNTAX  INTEGER {
                    disabled (1),
                    enabled  (2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Defines whether the Billing System is to perform a set of
                carrier-specific actions at the time given by
                nodeBillingDPTime.  This object can be used to schedule the
                generation of statistics on a 24-hour basis, for example."
        ::= {nodeBillingEntry 7 }

nodeBillingDPTime OBJECT-TYPE
        SYNTAX  TimeTicks
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Defines the time (given by the number of seconds since
                00:00:00 UCT) that the Billing System is to perform a set
                of carrier-specific actions.  This object can be used to
                schedule the generation of statistics on a 24-hour basis,
                for example."
        ::= {nodeBillingEntry 8 }

nodeBillingUsageRecOvflWarnings OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Count of the number of usage record counter-value overflow
                conditions that have occurred during the current
                aggregation period.  An overflow condition exists when an
                attempt was made to update a usage record counter, but
                such an update would have overflowed the counter.  In this
                case, the usage record is closed and a new one is opened,
                if there is sufficient space in the service's aggregated
                usage data store."
        ::= {nodeBillingEntry 9 }
```

```
nodeBillingTotalUsageRecOvflWarnings OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS      mandatory
        DESCRIPTION
                "Count of the number of usage record overflow conditions
                that have occurred during the current day."
        ::= {nodeBillingEntry 10}

nodeBillingBillableUsageEvents OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Count of the number of data units (e.g, SMDS L3 PDUs)
                processed for billing treatment."
        ::= {nodeBillingEntry 11 }

nodeBillingNonBillableUsageEvents OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Count of the number of data units (e.g, SMDS L3 PDUs) not
                considered for billing treatment."
        ::= {nodeBillingEntry 12 }

nodeBillingUsageRecCreated OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Count of the number of usage records created during the
                current aggregation period."
        ::= {nodeBillingEntry 13 }

nodeBillingTotalUsageRecCreated OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Count of the total number of usage records created during
                the current day."
        ::= {nodeBillingEntry 14 }

nodeBillingUsageRecCrFailures OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total number of usage records that could not be created
                during the current aggregation period.  This counter is
                normally incremented when the usage data file is at
                capacity and no additional usage records can be added."
        ::= {nodeBillingEntry 15 }

nodeBillingTotalUsageRecCrFailures OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total number of usage records that could not be created
                during the current day."
        ::= {nodeBillingEntry 16 }

nodeBillingUsageRecSent OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total number of usage records that have been transferred
                to the Adjunct Billing Processor during the current
                aggregation period."
        ::= {nodeBillingEntry 17 }

nodeBillingTotalUsageRecSent OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total number of usage records that have been transferred
                to the Adjunct Billing Processor during the current day."
        ::= {nodeBillingEntry 18 }

nodeBillingUsageDataStoreFull OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of times the aggregated usage data store became
                full during the current aggregation period."
        ::= {nodeBillingEntry 19 }
```

```
nodeBillingTotalUsageDataStoreFull OBJECT-TYPE
        SYNTAX  Counter
        ACCESS      read-only
        STATUS  mandatory
        DESCRIPTION
                "Total number of times the aggregated usage data store
                became full during the current day."
        ::= {nodeBillingEntry 20 }

nodeBillingAdminAction OBJECT-TYPE
        SYNTAX  INTEGER {
                    invalid (1),
                    forceUpload (2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "This object defines a set of administrative actions that
                can be performed by the Billing System.

                forceUpload - Forces the current aggregation period to end
                and the service's aggregated usage data file to be
                uploaded to the Adjunct Processor.  A new aggregation
                period is then started.  This action can be requested when
                billing is enabled or disabled for the service.

                This object always returns invalid(1) when read."
        ::= {nodeBillingEntry 21 }

nodeRerouteAlg OBJECT-TYPE
        SYNTAX  INTEGER {
                negpos (0),
                negneg (1),
                pospos (2),
                disable (3)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The number of seconds delay between each reroute batch
                event."
        ::= { node 74 }

nodeADTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF NodeADEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A list of Authentication Domain entries."
        ::= { node 75 }

nodeADEntry OBJECT-TYPE
        SYNTAX  NodeADEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "An Authentication Domain entry contains information about
                authentication servers used by this node."
        INDEX   { nodeADIndex }
        ::= { nodeADTable 1 }

NodeADEntry ::=
        SEQUENCE {
                nodeADIndex
                    Index,
                nodeADType
                    INTEGER,
                nodeADAdminStatus
                    INTEGER,
                nodeADSecrets
                    OCTET STRING,
                nodeADSvr1Addr
                    IpAddress,
                nodeADSvr1Retrys
                    INTEGER,
                nodeADSvr1Timeout
                    INTEGER,
                nodeADSvr2Addr
                    IpAddress,
                nodeADSvr2Retrys
                    INTEGER,
                nodeADSvr2Timeout
                    INTEGER,
                nodeADSvr3Addr
                    IpAddress,
                nodeADSvr3Retrys
                    INTEGER,
                nodeADSvr3Timeout
                    INTEGER
                }
```

```
nodeADIndex OBJECT-TYPE
        SYNTAX  Index
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The Domain ID which maps to a Domain name."
        ::= { nodeADEntry 1 }

nodeADType OBJECT-TYPE
        SYNTAX  INTEGER {
                radius (1)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The Authentication Domain type."
        ::= { nodeADEntry 2 }

nodeADAdminStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                invalid (0),
                up (1),
                down (2),
                deleted (3)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The current state of this entry."
        ::= { nodeADEntry 3 }

nodeADSecrets OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Secrets (encryption keys) used by the switch and
                authentication server."
        ::= { nodeADEntry 4 }

nodeADSvr1Addr OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The IP address of the primary Authentication Server."
        ::= { nodeADEntry 5 }

nodeADSvr1Retrys OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The # of times to retry sending an authentication request
                to the primary server in the case of no-response."
        ::= { nodeADEntry 6 }

nodeADSvr1Timeout OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The # of seconds to wait between authentication request
                retries for the primary server."
        ::= { nodeADEntry 7 }

nodeADSvr2Addr OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The IP address of the secondary Authentication Server."
        ::= { nodeADEntry 8 }

nodeADSvr2Retrys OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The # of times to retry sending an authentication request
                to the secondary server in the case of no-response."
        ::= { nodeADEntry 9 }

nodeADSvr2Timeout OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The # of seconds to wait between authentication request
                retries for the secondary server."
        ::= { nodeADEntry 10 }

nodeADSvr3Addr OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The IP address of the tertiary Authentication Server."
        ::= { nodeADEntry 11 }
```

```
nodeADSvr3Retrys OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The # of times to retry sending an authentication request
                to the tertiary server in the case of no-response."
        ::= { nodeADEntry 12 }

nodeADSvr3Timeout OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The # of seconds to wait between authentication request
                retries for the tertiary server."
        ::= { nodeADEntry 13 }

nodeOamAlarmDisabled OBJECT-TYPE
        SYNTAX  INTEGER {
                enabled(1),
                disabled(2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "If disabled then don't generate oam alarms for circuits
                that are down on this switch."
        ::= { node 76 }

nodeRefclocksrcTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF NodeRefclocksrcEntry
        ACCESS  not-accessible
        STATUS   mandatory
        DESCRIPTION
                "A list of generators (sources) for the system reference
                clock on the node. This reference clock is used by the
                clock generation unit (CGU) to create the system clock. To
                this reference clock the constant bit rate ATM interface
                cards (CBR cards) are synchronized."
        ::= { node 77 }
```

Table of Reference Clock Sources

```
nodeRefclocksrcEntry OBJECT-TYPE
        SYNTAX  NodeRefclocksrcEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "Information of a single system reference clock source."
        INDEX   { nodeRefclocksrcIndex }
        ::= { nodeRefclocksrcTable 1 }

NodeRefclocksrcEntry ::=
        SEQUENCE {
                nodeRefclocksrcIndex
                    INTEGER,
                nodeRefclocksrcPriority
                    INTEGER,
                nodeRefclocksrcType
                    INTEGER,
                nodeRefclocksrcSlotId
                    INTEGER,
                nodeRefclocksrcPportId
                    INTEGER
                }

nodeRefclocksrcIndex OBJECT-TYPE
        SYNTAX  INTEGER (1..20)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "A positive integer to identify an entry in the
                nodeRefclocksrcTable."
        ::= { nodeRefclocksrcEntry 1 }

nodeRefclocksrcPriority OBJECT-TYPE
        SYNTAX  INTEGER (1..20)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The priority of the system reference clock source.  A
                source with a nodeRefclocksrcPriority=n means that this
                source is used before a source with a higher value of
                nodeRefclocksrcPriority as long as this source is
                available. The system chooses one of the sources with
                the same lowest available value of
                nodeRefclocksrcPriority."
        ::= { nodeRefclocksrcEntry 2 }

nodeRefclocksrcType OBJECT-TYPE
        SYNTAX  INTEGER {
                    external (1),          -- external clock source
                    pport (2)              -- recovered from line interface
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The type of the reference clock source. It is either an
                external clock source connected to a ATM-IWU or DS3 card,
                or the reference clock is recovered from the line
                interface."
        ::= { nodeRefclocksrcEntry 3 }
```

```
nodeRefclocksrcSlotId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The card number of the reference clock source.  In case
                of an external clock interface this is the number of the
                card as defined in cardLogicalSlotId. In the case of a
                clock recovered from the line interface it is the number
                of the corresponding card as defined in cardLogicalSlotId."
        ::= { nodeRefclocksrcEntry 4 }

nodeRefclocksrcPportId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The interface number of the reference clock source.
                In case of an external clock interface this is meaningless.
                In case of a clock recovered from the line interface it is
                the number of the corresponding pport as defined in
                pportId."
        ::= { nodeRefclocksrcEntry 5 }

nodeRefclockActiveSrc OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The entry number of the active reference clock source in
                nodeRefclocksrcTable. If zero, the CGU is in free-running
                mode or holdover mode using the last PLL parameters."
        ::= { node 78 }

nodeRefclockActiveCGUSlotId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The physical slot id of the current master CGU board."
        ::= { node 79 }

nodeRefclockActiveCGUMode OBJECT-TYPE
        SYNTAX  INTEGER {
                free-running (1),
                sync-to-reference-clock (2),
                holdover (3),
                extended-holdover (4)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The clock mode of the current master CGU board."
        ::= { node 80 }

nodeInitiateImageBackup OBJECT-TYPE
        SYNTAX  INTEGER {
                proceed (1)    -- proceed with the backup
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "When set, copy all of the runtime application images
                and configuration images to there corresponding backup
                locations on the hard disk."
        ::= { node 81 }

nodeImageBackupState OBJECT-TYPE
        SYNTAX  INTEGER {
                proceeding (1),-- Proceeding with the backup
                done       (2) -- Done with backup operation
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Provides the current state of the images backup operation."
        ::= { node 82 }

nodeInitiateImageRestore OBJECT-TYPE
        SYNTAX  INTEGER {
                proceed (1)    -- Proceed with the restore
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "When set, copy all of the runtime application images
                and configuration images from their BACKUP locations on
                the hard disk to there corresponding ACTIVE locations on
                the hard disk.  Once completed, reboot the entire node."
        ::= { node 83 }

nodeApplicationTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF NodeApplicationEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A list of Application Image Descriptions."
        ::= { node 84 }

nodeApplicationEntry OBJECT-TYPE
        SYNTAX  NodeApplicationEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "An Application Entry contains a description of the
                 application image, the version of the PRIMARY copy and
                 the version of the SECONDARY copy."
        INDEX   { nodeApplicationIndex }
        ::= { nodeApplicationTable 1 }
```

```
NodeApplicationEntry ::=
        SEQUENCE {
                nodeApplicationIndex
                    Index,
                nodeApplicationDescription
                    DisplayString,
                nodePrimaryVersion
                    DisplayString,
                nodeSecondaryVersion
                    DisplayString
        }

nodeApplicationIndex OBJECT-TYPE
        SYNTAX  Index
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The numeric index of application images.
                    1 - SP Runtime Application
                    2 - IOM Type A Runtime Application
                    3 - IOM Type B Runtime Application
                    4 - IOM Type C Runtime Application
                    5 - IOM Type D Runtime Application"
        ::= { nodeApplicationEntry 1 }

nodeApplicationDescription OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The description of this Application Image."
        ::= { nodeApplicationEntry 2 }

nodePrimaryVersion OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The version of the PRIMARY Application Image."
        ::= { nodeApplicationEntry 3 }

nodeSecondaryVersion OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The version of the SECONDARY Application Image."
        ::= { nodeApplicationEntry 4 }
```

```
nodePrimarySyncRefAdminStateOBJECT-TYPE
        SYNTAX  INTEGER {
                externala (1), -- T1/E1 Rate External Clock 1
                externalb (2), -- T1/E1 Rate External Clock 2
                portrefa (3),  -- IOM Port Reference Clock 1
                portrefb (4),  -- IOM Port Reference Clock 2
                internal  (5)  -- Internal Free Running Clock
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The requested primary clock synchronization reference
                 source.  Default is Internal."
        ::= { node 85 }

nodePrimarySyncRefOperationalStateOBJECT-TYPE
        SYNTAX  INTEGER {
                externala (1), -- T1/E1 Rate External Clock 1
                externalb (2), -- T1/E1 Rate External Clock 2
                portrefa (3),  -- IOM Port Reference Clock 1
                portrefb (4),  -- IOM Port Reference Clock 2
                internal   (5) -- Internal Free Running Clock
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The actual primary clock synchronization reference
                source."
        ::= { node 86 }

nodeSecondarySyncRefAdminStateOBJECT-TYPE
        SYNTAX  INTEGER {
                externala (1), -- T1/E1 Rate External Clock 1
                externalb (2), -- T1/E1 Rate External Clock 2
                portrefa (3),  -- IOM Port Reference Clock 1
                portrefb (4),  -- IOM Port Reference Clock 2
                internal (5)   -- Internal Free Running Clock
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The requested secondary clock synchronization reference
                 source.  Default is Internal."
        ::= { node 87 }
```

```
nodeSecondarySyncRefOperationalStateOBJECT-TYPE
        SYNTAX  INTEGER {
                externala (1), -- T1/E1 Rate External Clock 1
                externalb (2), -- T1/E1 Rate External Clock 2
                portrefa (3),  -- IOM Port Reference Clock 1
                portrefb (4),  -- IOM Port Reference Clock 2
                internal   (5) -- Internal Free Running Clock
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The actual secondary clock synchronization reference
                 source."
        ::= { node 88 }
nodePrimaryPLLOperationalStateOBJECT-TYPE
        SYNTAX  INTEGER {
                active (1),               -- In-lock and active as timing
                                             reference
                inactive-in-lock (2),     -- Not active but in-lock
                unusable (3)              -- Unusable state
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The current state of the primary synchronization PLL.
                The 'unusable' state indicates that at the instant the
                PLL state was sampled, the PLL output was not usable as
                a system timing reference (e.g. the PLL is configured
                to use an external reference and that reference is
                physically disconnected)."
        ::= { node 89 }

nodeSecondaryPLLOperationalStateOBJECT-TYPE
        SYNTAX  INTEGER {
                active (1),               -- In-lock and active as timing
                                             reference
                inactive-in-lock (2),     -- Not active but in-lock
                unusable   (3)            -- Unusable state
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The current state of the secondary synchronization PLL.
                The 'unusable' state indicates that at the instant the
                PLL state was sampled, the PLL output was not usable as
                a system timing reference (e.g. the PLL is configured
                to use an external reference and that reference is
                physically disconnected)."
        ::= { node 90 }
```

```
nodeExternalClockAOperationalStateOBJECT-TYPE
        SYNTAX  INTEGER {
                active (1),     -- Valid
                ais    (2),     -- Detected AIS condition
                los    (3),     -- Detected Loss Of Signal
                lof    (4)      -- Detected Loss of Frame
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The current state of the External Clock #1 reference."
        ::= { node 91 }

nodeExternalClockBOperationalStateOBJECT-TYPE
        SYNTAX  INTEGER {
                active (1),     -- Valid
                ais    (2),     -- Detected AIS condition
                los    (3),     -- Detected Loss Of Signal
                lof    (4)      -- Detected Loss Of Frame
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The current state of the External Clock #2 reference."
        ::= { node 92 }

nodePortClockAOperationalState OBJECT-TYPE
        SYNTAX  INTEGER {
                active (1),     -- Valid
                down   (2)      -- Invalid
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The current state of the Port Clock A reference."
        ::= { node 93 }

nodePortClockBOperationalState OBJECT-TYPE
        SYNTAX  INTEGER {
                active (1),     -- Valid
                down   (2)      -- Invalid
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The current state of the Port Clock B reference."
        ::= { node 94 }
```

```
nodeExternalTimingSourceOBJECT-TYPE
        SYNTAX  INTEGER {
                primary (1),    -- external clock out derived from primary
                                   reference
                secondary (2), -- ext. clk. out is derived from secondary
                                   reference
                loopback-ext1 (3), -- ext. clk. out is loopback version of
                                      ext clk ref #1
                tx-ais (4)      -- ext. clk. out is unreferenced and
                                   Transmits AIS
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The state of the external timing output.  It may
                be derived from the primary reference or secondary
                reference, or it may be a loopbacked version of the
                external clock #1 input for test purposes.  In the
                tx-ais configuration the external clock out transmits
                an AIS indication continuously."
        ::= { node 95 }

nodeSyncAutoRestoreOBJECT-TYPE
        SYNTAX  INTEGER {
                enable (1),     -- Allow auto-restore
                disable(2)      -- Do not allow auto-restore
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Revertive state of system timing clock reference.
                When enabled, fallback from Primary to Secondary will
                revert back to Primary upon its recovery."
        ::= { node 96 }

nodeExternalClockInterfaceTypeOBJECT-TYPE
        SYNTAX  INTEGER {
                t1    (1),      -- T1 rate W/W terminals
                e1bnc (2),      -- E1 rate 75 ohm BNC
                e1ww  (3)       -- E1 rate W/W terminals
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The interface type of the External Clock references
                and External Clock Output."
        ::= { node 97 }

nodeTrapMaskMib OBJECT-TYPE
        SYNTAX OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The suppression mask for MIB II traps."
        ::= { node 98 }

nodeTrapMaskEnterprise OBJECT-TYPE
        SYNTAX OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The suppression mask for enterprise traps."
        ::= { node 99 }

nodeExternalClockOutLBO  OBJECT-TYPE
        SYNTAX INTEGER {
                len-zero-133ft (1),  -- length 0-133ft
                len-133-266ft (2),   -- 133-266ft
                len-266-399ft (3),   -- 266-399ft
                len-399-533ft (4),   -- 399-533ft
                len-533-655ft (5)    -- 533-655ft
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The line-build-out for the T1 External Clock
                Output from the switch.  Default is 0-133ft."
        ::= { node 100 }

nodeActiveTimingRefAdmin OBJECT-TYPE
        SYNTAX INTEGER {
                primary (1),  -- use primary PLL to drive system timing
                                 reference
                secondary (2)  -- use secondary PLL to drive system timing
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This variable controls the indication sent to
                the IOMs as to which of the two clock references
                is preferred.  Normally, it is left on the primary
                setting.  If the primary fails then the secondary
                automatically becomes the preferred source indicated
                to the IOMs.  This variable allows the operator to
                force the indication to secondary for test purposes."
        ::= { node 101 }
```

```
nodeTimingManualRestore OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "When this variable is written, it causes the
                system timing hardware to attempt to revert back
                to the previously-failed-but-since-recovered
                primary reference."
        ::= { node 102 }

nodeLanIpMask OBJECT-TYPE
        SYNTAX IpAddress
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The inband (Ethernet) interface IP mask.
                 the default is 255.255.255.0"
        ::= { node 103 }

nodeBulkStatsCollectorAddress OBJECT-TYPE
        SYNTAX IpAddress
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The IP Address of the Bulk Statistics Collector to which
                bulk statistics data is transferred."
        ::= { node 104 }

nodeBulkStatsTransferState OBJECT-TYPE
        SYNTAX INTEGER {
                inactive (1),
                pending  (2),
                active   (3),
                failed   (4)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object defines the state of a Bulk Statistics raw
                file transfer process to the Bulk Statistics Collector
                identified by the nodeBulkStatsCollector object. The
                possible values are:
                    inactive - No file is currently being transferred.
                    pending  - A file is awaiting transfer.
                    active   - A file is currently being transferred.
                    failed   - The last file transfer attempt has failed
                               and is being retried.
  "
                ::= { node 105 }

nodeAuthState OBJECT-TYPE
        SYNTAX  INTEGER {
                    auth-disabled (0),
                    auth-enabled (1)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Console Login Authentication for this node, enabled or
                 disabled."
        ::= { node 106 }

nodeAuthDomainID OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Authentication Domain ID for this node."
        ::= { node 107 }

nodeAuthFailReason OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Reason for console login authentication failure."
        ::= { node 108 }

nodeAuthLoginUser OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Login user failed to pass console authentication."
        ::= { node 109 }

nodeBulkAvgTransportBwUsed OBJECT-TYPE
        SYNTAX Gauge
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
               "This object reports the average amount of bandwidth (in
               bits per second) that has been used to transport ATM
               Bulk Statistics data to the Adjunct Processor during the
               current day."
        ::= { node 110 }
```

```
nodeBulkAvgTransportBwBurst OBJECT-TYPE
        SYNTAX Gauge
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "This object reports the average transport bandwidth burst
                rate (in bits per second) obtained to transport ATM Bulk
                Statistics data to the Adjunct Processor during the current
                day."
         ::= { node 111 }

nodeBulkMinTransportBwBurst OBJECT-TYPE
        SYNTAX Gauge
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "This object reports the minimum transport bandwidth burst
                rate (in bits per second) obtained to transport ATM Bulk
                Statistics data to the Adjunct Processor during the current
                day."
        ::= { node 112 }

nodeBulkMaxTransportBwBurst OBJECT-TYPE
        SYNTAX Gauge
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "This object reports the maximum transport bandwidth burst
                rate (in bits per second) obtained to transport ATM Bulk
                Statistics data to the Adjunct Processor during the current
                day."
        ::= { node 113 }

nodeBulkSwAPCommsFailures OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The number of times that communication from the switch to
                the ATM Bulk Statistics Adjunct Processor has failed
                during the current day.  A failure signifies failure of a
                file transfer operation to the Adjunct Processor."
        ::= { node 114 }

nodePsCStatus OBJECT-TYPE
        SYNTAX INTEGER {
                up (1),
                down (2),
                marginal (4)
        }
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The status of the power supply #3.  For a detailed
                explanation for the power supply being either in the down
                or marginal state, refer to the nodePsCDiagCode object."
         ::= { node 115}

nodePsCDiagCode OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The failure code of the power supply #3.  A non-zero value
                indicates one or more failures.  Refer to the hardware
                manual for a description of the failure code."
        ::= { node 116 }

nodeAdminStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                   notnplus1 (1),       -- Two Power Supplies
                   nplus1 (2)           -- Three Power Supplies
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "THIS OID IS OBSOLETE. THIS CAN BE REUSED.
                The desired setting of the  switch type indicating
                whether the   switch has two   power supplies
               (NotNplus1) or three (Nplus1)."
        ::= { node 117}

nodeOperatingStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                    twoPS (1),
                    threePS (2)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The actual  switch type indicating  whether the  switch
                has two power supplies (TwoPS)  or three (ThreePS)."
        ::= { node 118}
```

```
nodeTrapSeverity OBJECT-TYPE
        SYNTAX  INTEGER {
                    critical (1),
                    major (2),
                    minor (3),
                    nonalarm (4)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This is an OID placeholder for Trap Variables.  All
                Cascade traps will contain this binding for the severity
                level of the trap."
        ::= { node 119}

nodeTrapSequenceNumber OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This is an OID placeholder for Trap Variables.  All
                Cascade traps will contain this binding for the sequence
                number of the trap.  This is used to support more reliable
               traps."
        ::= { node 120}

nodeTrapTxRate OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Defines the number of traps (per second) that the switch
               is allowed to transmit"
       ::= { node 121}

nodeTrapMaskSeverity OBJECT-TYPE
       SYNTAX OCTET STRING
       ACCESS read-write
       STATUS mandatory
       DESCRIPTION
               "The suppression mask for trap severity levels.  See
               nodeTrapSeverity."
       ::= { node 122 }

nodeAlarmRelayStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                    activate (1),
                    deactive (2),
                    reset (3)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Control of the external CO alarm relays on the SPA."
        ::= { node 123}

nodeTrapsDiscardedMajor OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of Major severity traps that were discarded
                due to queue full conditions."
        ::= { node 124}

nodeTrapsDiscardedMinor OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
               "The number of Minor severity traps that were discarded
               due to queue full conditions."
        ::= { node 125}

nodeTrapDiscardedNonalarm OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of Nonalarm severity traps that were discarded
  due to queue full conditions."
       ::= { node 126}

nodeNtpExternalPoll OBJECT-TYPE
       SYNTAX INTEGER
       ACCESS read-write
       STATUS mandatory
       DESCRIPTION
               "The log base2 polling interval at which the active CP/SP
               shall request a time reference from each external reference
               time server, Range 6-10"
 ::= { node 127 }
```

```
nodeNtpLocalPoll OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The log base2 polling interval at which the IOP/IOM cards
                and redundant CP/SP shall request a time reference from
                each reference time server, Range 6-10"
 ::= { node 128 }

nodeNtpPreferredServer OBJECT-TYPE
        SYNTAX IpAddress
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The IP Address of the preferred reference time server"
 ::= { node 129 }

nodeNtpSystemStatus OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The Network Timing Protocol system summary status word
                Leap Year Indicator0 no warning
                   1 last minute has 61 sec
                   2 last minute has 59 sec
                   3 alarm condition
                  (clock not synchronized)"
 ::= { node 130 }

nodeNtpSystemEventCount OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "System Event Counter - exceptions since last read
                freezes at 15"
        ::= { node 131 }

nodeNtpSystemEventCode OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "System Event Code - Last Exception Code Reported"
        ::= { node 132 }

nodeNtpRefTimestampISec OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS  mandatory
        DESCRIPTION
                "Seconds portion of last Ntp clock update"
        ::= { node 133 }

nodeNtpRefTimeStampFSec OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS  mandatory
        DESCRIPTION
                "Fractions of a second of last Ntp clock update"
        ::= { node 134 }

nodeNtpOffset OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS  mandatory
        DESCRIPTION
                "Deviation of last Ntp clock update in msec."
 ::= { node 135 }

nodeNtpMaxOffset OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS  mandatory
        DESCRIPTION
                "Maximum Deviation between the Ntp clock msec.
                This value shall be cleared after it is read."
        ::= { node 136 }

nodeNtpNumberofUpdates OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS  mandatory
        DESCRIPTION
                "Number of time the local ntp timer clock has been updated
                by the NTP process since startup."
        ::= { node 137 }

nodeNtpReferenceID OBJECT-TYPE
        SYNTAX IpAddress
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This is the IP Address of the current time server
                being referenced"
        ::= { node 138 }
```

```
nodeBulkStatsBaseCollectPeriod OBJECT-TYPE
        SYNTAX INTEGER (0..60)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Defines the base collection period for Bulk Statistics
                on the B-STDX 8000/9000 starting with release 4.2.
                Only the following values are allowed: 0, 5, 15, 20, 30,
                60.The default value is 60 minutes.
                A value of 0 will disable Bulk Statistics."
        ::= { node 139 }

nodeLanIdleTimeout OBJECT-TYPE
        SYNTAX INTEGER (0..1440)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The idle timeout interval, in minutes, for the
                Ethernet interface.  If the Ethernet interface
                receives no valid IP traffic during this period,
                the interface is marked as idle and will not be
                used for outbound traffic.  Receipt of a valid
                IP frame restarts the idle timeout counter and
                reactivates the interface, if idle.
                Setting this value to 0 disables the idle timeout
                mechanism.  The default value is 10 minutes."
                DEFVAL { 10 }
        ::= { node 140 }
```

The Card Group

```
--  The variables that manage intelligent cards (PP's, CP's, IOP's).
--  For redundant pairs of cards, only the active card is managed.
--
--  NOTE: The card group is currently only supported in the B-STDX 9000
--  platform.

cardNumber OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of manageable cards physically present in this
                node. Standby cards are not considered manageable."
        ::= { card 1}

cardTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF CardEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A list of cardEntry's. The number of entries is given by
                the value of cardNumber"
        ::= { card 2}

cardEntry OBJECT-TYPE
        SYNTAX  CardEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "The card entry contains objects relevant to managing
                intelligent cards."
        INDEX       { cardPhysicalSlotId, cardRedundState }
        ::= { cardTable 1}

CardEntry  ::=
        SEQUENCE {
                cardLogicalSlotId
                    INTEGER,
                 cardPhysicalSlotId
                    INTEGER,
                cardAdminType
                    INTEGER,
                cardOperType
                    INTEGER,
                cardState
                    INTEGER,
                cardAdminStatus
                    INTEGER,
                cardOperStatus
                    INTEGER,
                cardDiagStatus
                    INTEGER,
                cardRedundConfig
                    INTEGER,
                cardRedundSlotMask
                    INTEGER,
                cardRedundActual
                    INTEGER,
                cardRedundState
                    INTEGER,
                cardToRedundant
                    TimeTicks,
                cardDiagNonFatalSource
                    INTEGER,
                cardDiagNonFatalTime
                    TimeTicks,
                cardDiagNonFatalErrMajor
                    INTEGER,
                cardDiagNonFatalErrMinor
                    INTEGER,
                cardDiagNonFatalStr
                    DisplayString,
                cardDiagFatalSource
                    INTEGER,
                cardDiagFatalTime
                    TimeTicks,
                cardDiagFatalErrMajor
                    INTEGER,
```

```
cardDiagFatalErrMinor
    INTEGER,
cardDiagFatalStr
    DisplayString,
cardDiagFatalReboots
    Counter,
cardDiagFatalAddress
    INTEGER,
cardDiagBackgroundPasses
    Counter,
cardDiagBackgroundFailures
    Counter,
cardDiagBackgroundSuccesses
    Counter,
cardDiagLEDReset
    INTEGER,
cardDiagPowerExtensive
    INTEGER,
cardCpuUtil
    Gauge,
cardMemoryUsage
    Gauge,
cardMaxVCs
    Gauge,
cardInUseVCs
    Gauge,
cardFreeVCs
    Gauge,
cardInOctets
    Counter,
cardInPkts
    Counter,
cardOutOctets
    Counter,
cardOutPkts
    Counter,
cardToWarmboot
    TimeTicks,
cardToColdboot
    TimeTicks,
cardModel
    DisplayString,
cardSerial
    OCTET STRING,
cardSwRev
    DisplayString,
cardHwRev
    DisplayString,
cardEpromRev
    DisplayString,
cardName
    DisplayString,
cardCktGroupTrap
    OCTET STRING,
cardOutBtus
    Counter,
cardInGoodBtus
    Counter,
cardInErrorBtus
    Counter,
cardInNoVcBtus
    Counter,
cardInLinkDownBtus
    Counter,
cardInNoBufferBtus
    Counter,
cardInForwardBitBtus
    Counter,
cardDiagTestId
    INTEGER,
cardDiagTestRuns
    INTEGER,
cardDiagState
    INTEGER,
cardDiagOptionStr
    OCTET STRING,
cardDiagPasses
    Counter,
cardDiagFailures
    Counter,
cardDiagResultString
    DisplayString,
cardFrameMemoryUtil
    Gauge,
cardResetPram
    INTEGER,
cardMemoryUtil
    Gauge,
cardFrameMemoryUsage
    Gauge,
cardUpTime
    TimeTicks,
cardPramChecksum
    INTEGER,
cardPhysicalIndex
    INTEGER,
cardExternalClockRate
    INTEGER,
cardShootState
    INTEGER,
cardEraseAll
    INTEGER,
cardAdminCapability
    INTEGER,
cardOperCapability
    INTEGER,
cardISDNswtype
    INTEGER,
```

```
cardCpuFgUtil
     Gauge,
cardTrkProtState
    INTEGER,
cardISDNSigType
    INTEGER,
cardISDNChanId
    INTEGER,
cardTransmitClockConfig
    INTEGER,
cardTransmitClockSwitchOver
    INTEGER,
cardTransmitClockStatus
    INTEGER,
cardSystemPrimaryClockPortConfig
    INTEGER,
cardSystemPrimaryClockStatus
    INTEGER,
cardSystemSecondaryClockPortConfig
    INTEGER,
cardSystemSecondaryClockStatus
    INTEGER,
cardInCells
    Counter,
cardInErrorCells
    Counter,
cardInErrorVPIVCI
    Counter,
cardOutCells
    Counter,
cardOutDiscardCells
    Counter,
cardQOSQueueSize
    INTEGER,
cardLastErrorPort
    INTEGER,
cardLastErrorVPI
    INTEGER,
cardLastErrorVCI
    INTEGER,
cardSystemPrimaryClockModeConfig
    INTEGER,
cardSystemSecondaryClockModeConfig
    INTEGER,
cardNFBDEStatus
    INTEGER,
cardProductCode
    DisplayString,
cardMfgPN
    DisplayString,
cardTotalUpTime
    TimeTicks,
cardIOAType
    INTEGER,
cardIOAHwRev
    DisplayString,
cardIOASerial
    DisplayString,
cardIOAProductCode
    DisplayString,
cardIOAMfgPN
    DisplayString,
cardDS0Support
    INTEGER,
cardDiagParamId
    INTEGER,
cardDiagParamValue
    INTEGER,
cardBulkStatsPeakCapability
    INTEGER,
cardBulkStatsTotalCapability
    INTEGER,
cardBulkStatsPeakEnable
    INTEGER,
cardBulkStatsTotalEnable
    INTEGER,
cardBulkStatsBaseCollectPeriod
    INTEGER,
cardNrtsHwRev
    INTEGER,
cardNrtsOutCellBufSize
    INTEGER,
cardNrtsOperState
    INTEGER,
cardNrtsAdminState
    INTEGER,
cardNrtsCcrmProtocolId
    INTEGER,
cardNrtsBcmProtocolId
    INTEGER,
cardNrtsRmGenInterval
    INTEGER,
cardNrtsIdleCktThresh
    INTEGER,
cardNrtsVbrNrtManage
    INTEGER,
cardNrtsIcrFact
    INTEGER,
cardNrtsMcastDiscardThresh
    INTEGER,
cardNrtsMcastDiscardCount
    Counter,
cardAdminIOAType
    INTEGER
}
```

```
cardLogicalSlotId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The logical slot number of this card. This is used as an
                index for the cardTable. In redundant configurations, this
                can be the physical slot number of either redundant card.
                In non redundant configurations, this is the same as
                cardPhysicalSlotId."
        ::= { cardEntry 1}

cardPhysicalSlotId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The physical slot number of this card. This can be differ-
                ent from cardLogicalSlotId in redundant configurations."
        ::= { cardEntry 2}

CardAdminType OBJECT-TYPE
        SYNTAX  INTEGER {
                    invalid (0),
                    v35-6 (1),              -- 6-port V.35 i/o card
                    ft1-1-24 (2),           -- 1-port 24-channel Fractional
                                               T1
                    fe1-1-30 (3),           -- 1-port 30-channel Fractional
                                               E1
                    uio-6 (4),              -- 6-port universal i/o card
                    cp1 (5),                -- Control Processor
                    uio-8 (6),              -- 8-port V.35 i/o card
                    ft1-4-24 (7),           -- 4-port 24-channel Fractional
                                               T1
                    fe1-4-30 (8),           -- 4-port 30-channel Fractional
                                               E1
                    ft3-1 (9),              -- 1-port Fractional T3 i/o card
                    fe3-1 (10),             -- 1-port Fractional E3 i/o card
                    hssi-1 (11),            -- HSSI i/o card
                    dsx1-10 (12),           -- 10-port DSX-1 card
                    rs232-18 (13),          -- 18-port X.21/V.24 I/O card,
                                               for STDX 3000/6000 only
                    rs232-8 (14),           -- 8-port X.21/V.24 I/O card,
                                               for STDX 3000/6000 only
                    ut1-4-24 (15),          -- 4-port 24-channel Un-
                                               Channelized T1
                    ue1-4-30 (16),          -- 4-port 30-channel Un-
                                               Channelized E1
                    atmds3-1 (17),          -- 1-port ATM DS3 UNI I/O card
                    atme3-1 (18),           -- 1-port ATM E3 UNI I/O card
                    pri-4 (19),             -- 4-port ISDN PRI I/O card
                    e1-pri-4 (20),          -- 4-port E1 PRI I/O card
                    sft1-4-24 (21),         -- 4-port short haul 24-channel
                                               Fractional T1 card
                    sut1-4-24 (22),         -- 4-port short haul 24-channel
                                               Un-Channelized T1 card
                    st1-pri-4 (23),         -- 4-port short haul PRI I/O card
                    t1-atm       (24),      -- T1 ATM
                    e1-atm       (25),      -- E1 ATM
                    ads3-t3   (26),         -- ATM DS3 T3 (topaz)
                    ads3-e3   (27),         -- ATM DS3 E3 (topaz)
                    cbr-ds1-s-4 (28),       -- 4-port CBR T1 structured card
                    cbr-ds1-us-4 (29),      -- 4-port CE T1 card
                    cbr-e1-s-4 (30),        -- 4-port CBR E1 structured card
                    cbr-e1-us-4 (31),       -- 4-port CE E1 card
                    atmiwu-1 (32),          -- 1-port ATM-IWU STM-1/STS-3c
                                               card
                    toc3-atm-4 (33),        -- 4-port Topaz OC3c ATM card
                    tstm1-atm-4 (34),       -- 4-port Topaz STM1 ATM card
                    sp-4       (35),        -- 4x4 switching processor
                    sp-8       (36),        -- 8x8 switching processor
                    atmcs-1     (37),       -- 1-port ATM-CS card (siemens)
                    toc12-atm-1 (38),       -- 1-port Topaz OC12c ATM card
                    tstm4-atm-1 (39),       -- 1-port Topaz STM4 ATM card
                    ads1-t1 (40),           -- Topaz 8 port T1 ATM
                    ads1-e1 (41),           -- Topaz 8 port E1 ATM
                    ads1-j2 (42)            -- Topaz 8 port J2 ATM
                    }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The desired card type."
        ::= { cardEntry 3}
```

```
cardOperType OBJECT-TYPE
SYNTAX  INTEGER {
                invalid (0),
                v35-6 (1),              -- 6-port V.35 i/o card
                ft1-1-24 (2),           -- 1-port 24-channel Fractional
                                        T1
                fe1-1-30 (3),           -- 1-port 30-channel FractionalE1
                uio-6 (4),              -- 6-port universal i/o card
                cp1 (5),                -- Control Processor
                uio-8 (6),              -- 8-port V.35 i/o card
                ft1-4-24 (7),           -- 4-port 24-channel Fractional
                                           T1
                fe1-4-30 (8),           -- 4-port 30-channel Fractional
                                           E1
                ft3-1 (9),              -- 1-port Fractional T3 i/o card
                fe3-1 (10),             -- 1-port Fractional E3 i/o card
                hssi-2 (11),            -- HSSI i/o card
                dsx1-10 (12),           -- 10-port DSX-1 card
                rs232-18 (13),          -- 18-port X.21/V.24 I/O card,
                                           for STDX 3000/6000 only
                rs232-8 (14),           -- 8-port X.21/V.24 I/O card,
                                           for STDX 3000/6000 only
                ut1-4-24 (15),          -- 4-port 24-channel Un-
                                           Channelized T1
                ue1-4-30 (16),          -- 4-port 30-channel Un-
                                           Channelized E1
                atmds3-1 (17),          -- 1-port ATM DS3 UNI I/O card
                atme3-1 (18),           -- 1-port ATM E3 UNI I/O card
                pri-4 (19),             -- 4-port ISDN PRI I/O card
                e1-pri-4 (20),          -- 4-port E1 PRI I/O card
                sft1-4-24 (21), -       - 4-port short haul 24-channel
                                          Fractional T1 card
                sut1-4-24 (22),         -- 4-port short haul 24-channel
                                           Un-Channelized T1 card
                st1-pri-4 (23),         -- 4-port short haul PRI I/O card
                t1-atm       (24),      -- T1 ATM
                e1-atm       (25),      -- E1 ATM
                ads3-t3   (26),         -- ATM DS3 T3 (topaz)
                ads3-e3   (27),         -- ATM DS3 E3 (topaz)
                cbr-ds1-s-4 (28),       -- 4-port CBR T1 structured card
                cbr-ds1-us-4 (29),      -- 4-port CE T1 card
                cbr-e1-s-4 (30),        -- 4-port CBR E1 structured card
                cbr-e1-us-4 (31),       -- 4-port CE E1 card
                atmiwu-1 (32),          -- 1-port ATM-IWU STM-1/STS-3c
                                           card
                toc3-atm-4 (33),        -- 4-port Topaz OC3c ATM card
                tstm1-atm-4 (34),       -- 4-port Topaz STM1 ATM card
                sp-4       (35),        -- 4x4 switching processor
                sp-8       (36),        -- 8x8 switching processor
                atmcs-1     (37),       -- 1-port ATM-CS card (siemens)
                toc12-atm-1 (38),       -- 1-port Topaz OC12c ATM card
                tstm4-atm-1 (39),       -- 1-port Topaz STM4 ATM card
                ads1-t1 (40),           -- Topaz 8 port T1 ATM
                ads1-e1 (41),           -- Topaz 8 port E1 ATM
                ads1-j2 (42)            -- Topaz 8 port J2 ATM
               }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The actual card type."
        ::= { cardEntry 4}

cardState OBJECT-TYPE
        SYNTAX  INTEGER {
                    unknown (0),
                    present (1),
                    loading (2),
                    start (3),
                    init (4),
                    sync (5),
                    syncdone (6),
                    ready (7),
                    active (8),
                    stopped (9),
                    down (10),
                    debug (11),
                    offlinediag (12)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The state of this card."
        ::= { cardEntry 5}

cardAdminStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                    invalid (0),
                    up (1),
                    down (2),
                    testing (3),
                    maintenance (6)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The desired status of this card."
        ::= { cardEntry 6}
```

```
cardOperStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                    invalid (0),
                    up (1),
                    down (2),
                    testing (3),
                    maintenance (6)
                }
        ACCESS  read-only
        STATUSmandatory
        DESCRIPTION
                "The current status of this card."
        ::= { cardEntry 7}

cardDiagStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                    unknown (0),
                    ok (1),
                    marginal (2)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The current diagnostics status of this card, as
                determined by background diagnostics."
        ::= { cardEntry 8}

cardRedundConfig OBJECT-TYPE
        SYNTAX  INTEGER {
                    none (0),
                    configured (1)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Is the card configured for redundancy?"
        ::= { cardEntry 9}

cardRedundSlotMask OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "A 32-bit mask indicating the slot position of this card
                and its redundant partner. The low bit is slot 1 and the
                high bit is slot32."
        ::= { cardEntry 10}

cardRedundActual OBJECT-TYPE
        SYNTAX  INTEGER {
                    connected (1),
                    not-connected (2)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Is the card currently connected to a redundant card?"
        ::= { cardEntry 11}

cardRedundState OBJECT-TYPE
        SYNTAX  INTEGER {
                    active (1),
                    standby (2)
                    }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The current redundancy state of this card.  Exception:
                when used as an index for the following objects:
                cardAdminType, cardAdminCapability, and cardAdminStatus,
                the value 1 indicatesthe card of the redundant pair in the
                lower-numbered slot, and the value 2 indicates the card in
                the higher-numbered slot."
        ::= { cardEntry 12}

cardToRedundant OBJECT-TYPE
        SYNTAX  TimeTicks
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Instructs the standby card to reset the active card and
                take over as new active card. The reset takes place after
                specified time ticks have elapsed. A value of 0 indicates
                cancellation of the previously scheduled request."
        ::= { cardEntry 13}
```

```
cardDiagNonFatalSource OBJECT-TYPE
        SYNTAX  INTEGER {
                power-on-diagnostics (1),
                background-diagnostics (2),
                fault (3),
                frame-heap (4),
                redundancy (5),
                system-level (6),
                card-level (7),
                i960-data-structures (8),
                general (9),
                data-alignment (10),
                device-driver-level (11)
        }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Source who reported last non-fatal error."
        ::= { cardEntry 14 }

cardDiagNonFatalTime OBJECT-TYPE
        SYNTAX  TimeTicks
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Time the last non-fatal error was reported."
        ::= { cardEntry 15 }

cardDiagNonFatalErrMajor OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Major error code of last non-fatal error."
        ::= { cardEntry 16 }

cardDiagNonFatalErrMinor OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Minor error code of last non-fatal error."
        ::= { cardEntry 17 }

cardDiagNonFatalStr OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Ascii string describing last non-fatal error."
        ::= { cardEntry 18 }

cardDiagFatalSource OBJECT-TYPE
        SYNTAX  INTEGER {
                power-on-diagnostics (1),
                background-diagnostics (2),
                fault (3),
                frame-heap (4),
                redundancy (5),
                system-level (6),
                card-level (7),
                i960-data-structures (8),
                general (9),
                data-alignment (10),
                device-driver-level (11)
        }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Source who reported last fatal error."
        ::= { cardEntry 19 }

cardDiagFatalTime OBJECT-TYPE
        SYNTAX  TimeTicks
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Time the last fatal error was reported."
        ::= { cardEntry 20 }

cardDiagFatalErrMajor OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Major error code of last fatal error."
        ::= { cardEntry 21 }

cardDiagFatalErrMinor OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Minor error code of last fatal error."
        ::= { cardEntry 22 }

cardDiagFatalStr OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Ascii string describing last fatal error."
        ::= { cardEntry 23 }
```

```
cardDiagFatalReboots OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Number of times the switch has re-booted since last
                fatal error was reported."
        ::= { cardEntry 24 }

cardDiagFatalAddress OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Address switch was executing when it encountered fatal
                error."
        ::= { cardEntry 25 }

cardDiagBackgroundPasses OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Number of passes made by the background diagnostics."
        ::= { cardEntry 26}

cardDiagBackgroundFailures OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Number of failures discovered by background diagnostics."
        ::= { cardEntry 27 }

cardDiagBackgroundSuccesses OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Number of successful passes by background diagnostics."
        ::= { cardEntry 28 }

cardDiagLEDReset OBJECT-TYPE
        SYNTAX  INTEGER {
                state-unchanged (0),
                state-to-active (1)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Change the card state from marginal to active.  Change the
                LED from yellow to green."
        ::= { cardEntry 29 }

cardDiagPowerExtensive OBJECT-TYPE
        SYNTAX  INTEGER {
                fast-tests (0),
                extensive-tests (1)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Should power on diagnostics do slower, extensive testing,
                or fast testing?"
        ::= { cardEntry 30 }

cardCpuUtil OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The CPU utilization percentage for packet processing on
                this intelligent card."
        ::= { cardEntry 31 }

cardMemoryUsage OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The memory utilization, in terms of free bytes, for
                this intelligent card."
        ::= { cardEntry 32 }

cardMaxVCs OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of available VC entries on this card."
        ::= { cardEntry 33 }

cardInUseVCs OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of currently in use VC entries on this card."
        ::= { cardEntry 34 }

cardFreeVCs OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of currently free VC entries on this card."
        ::= { cardEntry 35 }
```

```
cardInOctets OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of octets received on this card,
                including framing bytes."
        ::= { cardEntry 36 }

cardInPkts OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of packets received on this card."
        ::= { cardEntry 37 }

cardOutOctets OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of octets transmitted out of this card,
                including framing bytes."
        ::= { cardEntry 38 }

cardOutPkts OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of packets requested to be transmitted
                out of this card, including those that were discarded or
                not sent."
        ::= { cardEntry 39 }

cardToWarmboot OBJECT-TYPE
        SYNTAX  TimeTicks
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Warm boots the card after the specified time ticks have
                elapsed. A value of 0 indicates cancellation of the
                previously scheduled re-boot request."
        ::= { cardEntry 40 }

cardToColdboot OBJECT-TYPE
        SYNTAX  TimeTicks
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Cold boots the card after the specified time ticks have
                elapsed. A value of 0 indicates cancellation of the
                previously scheduled re-boot request."
        ::= { cardEntry 41 }

cardModel OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The part number of the card."
        ::= { cardEntry 42 }

cardSerial OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The serial number of the card."
        ::= { cardEntry 43 }

cardSwRev OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The software rev number (major.minor)."
        ::= { cardEntry 44 }

cardHwRev OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The hardware rev number (major.minor)."
        ::= { cardEntry 45 }

cardEpromRev OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The EPROM firmware rev number (major.minor)."
        ::= { cardEntry 46 }
```

```
cardName OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The name of this card"
        ::= { cardEntry 47 }

cardCktGroupTrap OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This is a ( 4 + n*134 ) byte octet string where n => 1.
                The first and second byte is the interface/slot id of the
                link/IOP going down link/IOP going down or link goes up.
                The third byte is the reason why circuits go down:
                    1 is a user link goes down,
                    2 is a trunk link goes down,
                    3 is an IOP goes down and
                    4 is an user link goes up
                The fourth byte is the count of entries in the following
                circuit bit map array. Each circuit bit map entry is 134
                bytes. The first 2 bytes are the IOP slot on which dlci's
                are defined. The second 2 bytes are the Physical port ID
                on which dlci's are defined. The third 2 bytes are the
                interface id on which dlci's are defined. The following 128
                bytes (1024 bits) is a bit map for dlci's defined on the
                interface.  The left-most bit of the first byte represents
                dlci 0 and the right-most bit of the 128th byte represents
                dlci 1023. If a bit is set, it means the corresponding
                dlci on that interface is up/down."
        ::= { cardEntry 48 }

cardOutBtus OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of bus transfer units sent by this card."
        ::= { cardEntry 49 }

cardInGoodBtus OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of good bus transfer units received by this
                card."
        ::= { cardEntry 50 }

cardInErrorBtus OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of bus transfer units received with errors."
        ::= { cardEntry 51 }

cardInNoVcBtus OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of bus transfer units discarded because no
                circuit was found."
        ::= { cardEntry 52 }

cardInLinkDownBtus OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of bus transfer units discarded because the
                outgoing link was inactive."
        ::= { cardEntry 53 }

cardInNoBufferBtus OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of bus transfer units discarded because no
                buffer was available."
        ::= { cardEntry 54 }

cardInForwardBitBtus OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of bus transfer units discarded because the
                circuit forward bit was off."
        ::= { cardEntry 55 }

cardDiagTestId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Identification of the diagnostics tests to be run."
        ::= { cardEntry 56 }
```

```
cardDiagTestRuns OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The number of passes of the diagnostics tests to be run.
                The default value is 1."
        ::= { cardEntry 57 }

cardDiagState OBJECT-TYPE
        SYNTAX  INTEGER {
                inactive (0),
                active(1)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The current state of the foreground diagnostics on this
                card."
        ::= { cardEntry 58 }

cardDiagOptionStr OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Optional parameters to the diagnostic."
        ::= { cardEntry 59 }

cardDiagPasses OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Number of successful diagnostic passes."
        ::= { cardEntry 60 }

cardDiagFailures OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Number of failed diagnostic passes."
        ::= { cardEntry 61 }

cardDiagResultString OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Description of last diagnostic failure."
        ::= { cardEntry 62 }

cardFrameMemoryUtil OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The percentage of frame memory utilization on this
                intelligent card."
        ::= { cardEntry 63 }

cardResetPram OBJECT-TYPE
        SYNTAX  INTEGER {
                reset-pram (1)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Setting this to 1 causes a reset of PRAM on the
                corresponding card."
        ::= { cardEntry 64 }

cardMemoryUtil OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The percentage of system memory utilization on this
                intelligent card."
        ::= { cardEntry 65 }

cardFrameMemoryUsage OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The frame memory utilization, in terms of free bytes, for
                this intelligent card."
        ::= { cardEntry 66 }
```

```
cardUpTime OBJECT-TYPE
        SYNTAX  TimeTicks
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The time (in hundreths of a second) since this card was
                last booted."
        ::= { cardEntry 67 }

cardPramChecksum OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The PRAM checksum for this card."
        ::= { cardEntry 68 }
cardPhysicalIndex OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The physical slot id of this card."
        ::= { cardEntry 69 }

cardExternalClockRate OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The external clock rate for DSX1, T1 and E1 cards.  The
                value is specified in units of 8KHz and can range from 1
                to 255."
        ::= { cardEntry 70 }

cardShootState OBJECT-TYPE
        SYNTAX  INTEGER {
                shoot-disabled (0),
                shoot-enabled  (1)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Specifies whether the system monitor on the CP should
                shoot this card if it's unreachable. The default is
                enabled. The card index used in the binding refers to a
                physical card."
        ::= { cardEntry 71 }

cardEraseAll OBJECT-TYPE
        SYNTAX  INTEGER {
                erase-all (1)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "When set to 1, erases program flash and PRAM on specified
                card."
        ::= { cardEntry 72 }

cardAdminCapability OBJECT-TYPE
        SYNTAX  INTEGER {
                invalid (0),
                iop-frame-relay (1),
                iop-multi-service (2),
                cp-basic (4),
                cp-plus (8)
        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The desired capability of the card"
        ::= { cardEntry 73 }

cardOperCapability OBJECT-TYPE
        SYNTAX  INTEGER {
                frame-relay (1),
                multi-service (2)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The actual capability of the card"
        ::= { cardEntry 74 }

cardISDNswtype OBJECT-TYPE
        SYNTAX  INTEGER {
                att-4ess (1),
                att-5ess (2),
                dms100(3)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Specifies the ISDN central office switch type for the
                card."
        ::= { cardEntry 75 }
```

```
cardCpuFgUtil OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The CPU foreground utilization percentage for packet
                processing on this intelligent card."
        ::= { cardEntry 76 }

cardTrkProtState OBJECT-TYPE
        SYNTAX  INTEGER {
                enabled (1),
                disabled (2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Control for issue of Link Trunk Protocol request frames.
                If set to disabled, the issue of trunk protocol requests is
                blocked for all trunks on this card."
        ::= { cardEntry 77 }

cardISDNSigType OBJECT-TYPE
        SYNTAX  INTEGER {
                no-nfas (1),
                nfas (2)
        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Specifies whether the ISDN Signalling type is Associated
                or Non-Facility Associated Signalling"
        ::= { cardEntry 78 }

cardISDNChanId OBJECT-TYPE
        SYNTAX  INTEGER {
                exclusive (1),
                preferred (2)
        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Specifies whether Channel ID assignment is performed by
                the central office switch or the 9000 - Exclusive (central
                office)
                Preferred (9000)"
        ::= { cardEntry 79 }

cardTransmitClockConfig OBJECT-TYPE
        SYNTAX  INTEGER {
                system-primary-clock (1),
                system-secondary-clock (2),
                system-primary-secondary-clock (3),
                free-running-clock (4)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The transmit clock source for timing section of the card"
        ::= { cardEntry 80 }

cardTransmitClockSwitchOver OBJECT-TYPE
        SYNTAX  INTEGER {
                enable (1),
                disable (2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Enable/disable automatic transmit clock source switch-over
                once a failed clock reference has recovered."
        ::= { cardEntry 81 }

cardTransmitClockStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                in-synchronization (1),
                loss-of-synchronization (2)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The transmit clock PLL synchronization status of the card"
        ::= { cardEntry 82 }

cardSystemPrimaryClockPortConfig OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The port number (1-x) on the card that specifies the
                system primary clock source. A zero specifies no port"
        ::= { cardEntry 83 }
```

```
cardSystemPrimaryClockStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                normal(1),
                failure(2)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The system primary clock status as detected on the card."
        ::= { cardEntry 84 }

cardSystemSecondaryClockPortConfig OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The port number (1 to x) on the card that specifies the
                system secondary clock source.  A zero specifies no port"
        ::= { cardEntry 85 }

cardSystemSecondaryClockStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                normal(1),
                failure(2)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The system secondary clock status as detected on the
                card."
        ::= { cardEntry 86}

cardInCells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of cells received on this card."
        ::= { cardEntry 87 }

cardInErrorCells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of cells received on this card with HEC
                error."
        ::= { cardEntry 88 }

cardInErrorVPIVCI OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of cells received with HEC invalid VPI,
                VCI."
        ::= { cardEntry 89 }

cardOutCells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of cells transmitted out of this card"
        ::= { cardEntry 90 }

cardOutDiscardCells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of cells discarded due to congestion on
                this card."
        ::= { cardEntry 91 }

cardQOSQueueSize OBJECT-TYPE
        SYNTAX  INTEGER {
                cell-8K (1),
                cell-24K (2)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The ATM QOS queue size (per cell) of each port on the
                card"
        ::= { cardEntry 92 }

cardLastErrorPort OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The port number (1 to x) of the last cell with invalid
                cell header on the card"
        ::= { cardEntry 93 }
```

```
cardLastErrorVPI OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The VPI of the last cell with invalid cell header on the
                card"
        ::= { cardEntry 94 }

cardLastErrorVCI OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The VCI of the last cell with invalid cell header on the
                card"
        ::= { cardEntry 95 }

cardSystemPrimaryClockModeConfig OBJECT-TYPE
        SYNTAX  INTEGER {
                line-rate (1),
                plcp-mode (2)
        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The system primary clock mode. PLCP is for DS3 port only"
        ::= { cardEntry 96 }

cardSystemSecondaryClockModeConfig OBJECT-TYPE
        SYNTAX  INTEGER {
                line-rate (1),
                plcp-mode (2)
        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The system secondary clock mode. PLCP is for DS3 port
                only"
        ::= { cardEntry 97 }

cardNFBDEStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                cleared (1),
                not-cleared (2)
        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Non Fatal Background Diagnostic Error Status"
        ::= { cardEntry 98 }

cardProductCode OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This card's product code."
        ::= { cardEntry 99 }

cardMfgPN OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This card's manufacturing part number."
        ::= { cardEntry 100 }

cardTotalUpTime OBJECT-TYPE
        SYNTAX  TimeTicks
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This card's total accumulated up time, in hours."
        ::= { cardEntry 101 }

cardIOAType OBJECT-TYPE
        SYNTAX INTEGER {
                spa (1),                    -- Topaz SPA, redundant
                toc3-4 (2),                 -- Topaz OC3/STM1 MM 4-port SC,
                                            non-redundant
                tstm1-4 (3),                -- Topaz STM-1 MM 4-port SC, non-
                                            redundant OBSOLETE, use (2
                tds3-8 (4),                 -- Topaz DS3 8-port, non-
                                            redundant
                te3-8 (5),                  -- Topaz E3 8-port, non-redundant
                toc3-smfir-n-4 (6),         -- Topaz OC3/STM1 SMF,
                                            intermediate reach, 4-port SC,
                                            non-redundant
                toc3-smfir-r-4 (7),         -- Topaz OC3/STM1 SMF,
                                            intermediate reach, 4-port SC,
                                            redundant
                toc3-mm-r-4 (8),            -- Topaz OC3/STM1 MM, 4-port SC,
                                            redundant\
                toc3-smflr-n-4 (9),         -- Topaz OC3/STM1 SMF, long
                                            reach, 4-port SC,non-redundant
                toc3-smflr-r-4 (10),        -- Topaz OC3/STM1 SMF, long
                                            reach, 4-port SC, redundant
                tstm1-smfir-n-4 (11),       -- Topaz STM-1 SMF, intermediate
                                            reach, 4-port SC, non-redundant
                                            OBSOLETE, use (6)
                tstm1-smfir-r-4 (12),       -- Topaz STM-1 SMF, intermediate
                                            reach, 4-port SC, redundant
                                            OBSOLETE, use (7)
                tstm1-mm-r-4 (13),           -- Topaz STM-1 MM, 4-port SC,
                                            redundant
```

```
                        OBSOLETE, use (8)
tstm1-smflr-n-4 (14),   -- Topaz STM-1 SMF, long reach,
                        4-port SC, non-redundant
                        OBSOLETE, use (9)
tstm1-smflr-r-4 (15),   -- Topaz STM-1 SMF, long reach,
                        4-port SC, redundant OBSOLETE,
                        use (10)
tds3-r-8 (16),          -- Topaz DS3 8-port, redundant
te3-r-8 (17),           -- Topaz E3 8-port, redundant
hssi-n (18),            -- BSTDX HSSI non-redundant IOA
hssi-r (19),            -- BSTDX HSSI redundant IOA
toc12-smf-n-1(21),      -- Topaz OC12/STM4 SMF non-
                        redundant IOA
tads1-t1-n-8 (22),      -- Topaz T1 ATM - T1 non-
                        redundant IOA
tads1-t1-r-8 (23),      -- Topaz T1 ATM - T1 redundant
                        IOA
tads1-e1-75-n-8 (24),   -- Topaz T1 ATM - E1 75 ohm non-
                        redund IOA
tads1-e1-75-r-8 (25),   -- Topaz T1 ATM - E1 75 ohm
                        redundant IOA
tads1-e1-120-n-8 (26),  -- Topaz T1 ATM - E1 120 ohm non-
                        red IOA
tads1-e1-120-r-8 (27),  -- Topaz T1 ATM - E1 120 ohm
                        redundant IOA
tads1-j2-75-n-8 (28),   -- Topaz T1 ATM - J2 75 ohm non-
                        red IOA
tads1-j2-75-r-8 (29),   -- Topaz T1 ATM - J2 75 ohm
                        redundant IOA
tads1-j2-120-n-8 (30),  -- Topaz T1 ATM - J2 120 ohm non-
                        red IOA
tads1-j2-120-r-8 (31),  -- Topaz T1 ATM - J2 120 ohm
                        redundant IOA
bCS-DS3-n-1 (35),       -- BSTDX DS3 (CS) non redundant
                        IOA
bCS-DS3-r-1 (36),       -- BSTDX DS3 (CS) redundant IOA
bIWU-OC3-mm-n-1 (37),   -- BSTDX OC3 (IWU) multimode non
                        redundant IOA
bIWU-OC3-smfir-n-1 (38), -- BSTDX OC3 (IWU) single mode
                        intermediate reach non-redundant
                        IOA
toc3-stm1copper-n-4 (39), -- Topaz OC3/STM1 Copper, 4-port
                        SC, non-redundant
toc3-stm1copper-r-4 (40), -- Topaz OC3/STM1 Copper, 4-port
                        SC, redundant
spa-universal (32),     -- Topaz Universal SPA (E1/T1)
nplus1-chassis (41),    -- N+1 Chassis Indicator
toc12-smflr-n-1(42),    -- Topaz OC12/STM4 SMF long
                        reach non redundant IOA
npa-universal (43),     -- Garnet Universal NPA
bIWU-OC3-bumm-1 (44),   -- BSTDX OC3 (IWU) base unit
                        multimode redundant IOA
bIWU-OC3-bumm-smfir-1 (45), -- BSTDX OC3 (IWU) base unit
                        single mode intermediate reach
                        redundant IOA
bIWU-OC3-trm-mm-1 (46), -- BSTDX
                        OC3 (IWU) tranceiver module
                        multimode redundant IOA
bIWU-OC3-trm-smfir-1 (47), -- BSTDX OC3 (IWU) tranceiver
                        module single mode intermediate
                        reach redundant IOA
uio-v35 (48),           -- BSTDX UIO-V35 IOA
uio-x21 (49),           -- BSTDX UIO-X21 IOA
bds1-e1-75-n-12 (50),   -- BSTDX 12 port - E1 75 ohm non-
                        edund IOA
bds1-e1-75-r-12 (51),   -- BSTDX 12 port - E1 75 ohm
                        redundant IOA
bds1-e1-120-n-12 (52),  -- BSTDX 12 port - E1 120 ohm
                        non-red IOA
bds1-e1-120-r-12 (53)   -- BSTDX 12 port - E1 120 ohm
                        redundant IOA

                }
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The type of IOA attached to this card."
 ::= { cardEntry 102 }

cardIOAHwRev OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The attached IOA's hardware revision."
        ::= { cardEntry 103 }

cardIOASerial OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The attached IOA's serial number."
        ::= { cardEntry 104 }

cardIOAProductCode OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The attached IOA's product code."
        ::= { cardEntry 105 }
```

```
cardIOAMfgPN OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The attached IOA's manufacturing part number."
        ::= { cardEntry 106 }

cardDS0Support OBJECT-TYPE
        SYNTAX  INTEGER {
                ds0-lpbk-not-supported   (1),
                ds0-lpbk-supported   (2)
        }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Some B-Stdx T1 boards do not support lpbk."
        ::= { cardEntry 107 }

cardDiagParamId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Parameter identifier for internal diagnostics."
        ::= { cardEntry 108 }
cardDiagParamValue OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Value assigned to parameter specified by cardDiagParamId."
        ::= { cardEntry 109 }

cardBulkStatsPeakCapability OBJECT-TYPE
        SYNTAX INTEGER {
                disabled (1),
                enabled (2)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Configures capability to collect peak 5-minute statistics.
                Capability change takes effect upon IOM reboot."
        ::= { cardEntry 110 }

cardBulkStatsTotalCapability OBJECT-TYPE
        SYNTAX INTEGER {
                disabled (1),
                enabled (2)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Configures capability to collect total statistics.
                Capability change takes effect upon IOM reboot."
        ::= { cardEntry 111 }

cardBulkStatsPeakEnable OBJECT-TYPE
        SYNTAX INTEGER {
                disabled (1),
                enabled (2)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Reflects the operational state of the peak 5-minute
                statistic collection on an IOM. For this setting to be
                enabled, the respective capability must be enabled."
        ::= { cardEntry 112 }

cardBulkStatsTotalEnable OBJECT-TYPE
        SYNTAX INTEGER {
                disabled (1),
                enabled (2)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Reflects the operational state of the total statistic
                collection on an IOM. For this setting to be enabled,
                the respective capability must be enabled."
        ::= { cardEntry 113 }

cardBulkStatsBaseCollectPeriod OBJECT-TYPE
        SYNTAX INTEGER (15..1440)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Configures base collection period for bulk statistics
                on an IOM in minutes. Only the following values are
                allowed: 15, 20, 30, 60, 120, 180, 240, 360, 720, 1440.
                The default value is 60."
        ::= { cardEntry 114 }
```

```
cardNrtsHwRev OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Hardware revision of the NRTS processor."
        ::= { cardEntry 115 }

cardNrtsOutCellBufSize OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The size of the NRTS output cell buffer."
        ::= { cardEntry 116 }

cardNrtsOperState OBJECT-TYPE
        SYNTAX INTEGER {
                absent (1),
                down (2),
                up (3)
        }
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The operational status of the NRTS processor."
        ::= { cardEntry 117 }

cardNrtsAdminState OBJECT-TYPE
        SYNTAX INTEGER {
                disabled (1),
                enabled (2)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The admin status of the NRTS processor."
                ::= { cardEntry 118 }

cardNrtsCcrmProtocolId OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The protocol ID field for the CCRM cells. Must be
                different from the BCM protocol ID. Default is 6."
        ::= { cardEntry 119 }

cardNrtsBcmProtocolId OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The protocol ID field for the BCM cells. Must be different
                from the CCRM protocol ID. Default is 5."
        ::= { cardEntry 120 }

cardNrtsRmGenInterval OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The RM cell generation interval in ms. Minimum is 30,
                maximum is 100, default is 100."
        ::= { cardEntry 121 }

cardNrtsIdleCktThresh OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The number of RM cell generation intervals with no cell
                reception upon which the circuit is declared idle.
                Minimum is 1, maximum is 8, default is 8."
        ::= { cardEntry 122 }

cardNrtsVbrNrtManage OBJECT-TYPE
        SYNTAX INTEGER {
                no (1),
                yes (2)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Indicates whether VBR NRT traffic should be managed by
                the NRTS processor."
        ::= { cardEntry 123 }

cardNrtsIcrFact OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "A factor used to calculate the Initial Cell Rate for a
                circuit from its MCR and PCR."
        ::= { cardEntry 124 }
```

```
cardNrtsMcastDiscardThresh OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "A NRTS discard threshold for the multicast circuits."
        ::= { cardEntry 125 }

cardNrtsMcastDiscardCount OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "A NRTS discard counter for the multicast circuits."
        ::= { cardEntry 126 }

cardAdminIOAType OBJECT-TYPE
        SYNTAX INTEGER
                spa (1),                -- Topaz SPA, redundant
                toc3-4 (2),             -- Topaz OC3/STM1 MM 4-port SC,
                                        non-redundant
                tstm1-4 (3),            -- Topaz STM-1 MM 4-port SC, non-
                                        redundant OBSOLETE, use (2
                tds3-8 (4),             -- Topaz DS3 8-port, non-
                                        redundant
                te3-8 (5),              -- Topaz E3 8-port, non-redundant
                toc3-smfir-n-4 (6),     -- Topaz OC3/STM1 SMF,
                                        intermediate reach, 4-port SC,
                                        non-redundant
                toc3-smfir-r-4 (7),     -- Topaz OC3/STM1 SMF,
                                        intermediate reach, 4-port SC,
                                        redundant
                toc3-mm-r-4 (8),        -- Topaz OC3/STM1 MM, 4-port SC,
                                        redundant
                toc3-smflr-n-4 (9),     -- Topaz OC3/STM1 SMF, long
                                        reach, 4-port SC, non-redundant
                toc3-smflr-r-4 (10),    -- Topaz OC3/STM1 SMF, long
                                        reach, 4-port SC, redundant
                tstm1-smfir-n-4 (11),   -- Topaz STM-1 SMF, intermediate
                                        reach, 4-port SC, non-redundant
                                        OBSOLETE, use (6)
                tstm1-smfir-r-4 (12),   -- Topaz STM-1 SMF, intermediate
                                        reach, 4-port SC, redundant
                                        OBSOLETE, use (7)
                tstm1-mm-r-4 (13),      -- Topaz STM-1 MM, 4-port SC,
                                        redundant
                                        OBSOLETE, use (8)
                tstm1-smflr-n-4 (14),   -- Topaz STM-1 SMF, long reach,
                                        4-port SC, non-redundant
                                        OBSOLETE use (9)
                tstm1-smflr-r-4 (15),   -- Topaz STM-1 SMF, long reach,
                                        4-port SC, redundant OBSOLETE,
                                        use (10)
                tds3-r-8 (16),          -- Topaz DS3 8-port, redundant
                te3-r-8 (17),           -- Topaz E3 8-port, redundant
                hssi-n (18),            -- BSTDX HSSI non-redundant IOA
                hssi-r (19),            -- BSTDX HSSI redundant IOA
                toc12-smf-n-1(21),      -- Topaz OC12/STM4 SMF non-
                                        redundant IOA
                tads1-t1-n-8 (22),      -- Topaz T1 ATM - T1 non-
                                        redundant IOA
                tads1-t1-r-8 (23),      -- Topaz T1 ATM - T1 redundant
                                        IOA
                tads1-e1-75-n-8 (24),   -- Topaz T1 ATM - E1 75 ohm non-
                                        redund IOA
                tads1-e1-75-r-8 (25),   -- Topaz T1 ATM - E1 75 ohm
                                        redundant IOA
                tads1-e1-120-n-8 (26),  -- Topaz T1 ATM - E1 120 ohm non-
                                        red IOA
                tads1-e1-120-r-8 (27),  -- Topaz T1 ATM - E1 120 ohm
                                        redundant IOA
                tads1-j2-75-n-8 (28),   -- Topaz T1 ATM - J2 75 ohm non-
                                        red IOA
                tads1-j2-75-r-8 (29),   -- Topaz T1 ATM - J2 75 ohm
                                        redundant IOA
                tads1-j2-120-n-8 (30),  -- Topaz T1 ATM - J2 120 ohm non-
                                        red IOA
                tads1-j2-120-r-8 (31),  -- Topaz T1 ATM - J2 120 ohm
                                        redundant IOA
                bCS-DS3-n-1 (35),       -- BSTDX DS3 (CS) non redundant
                                        IOA
                bCS-DS3-r-1 (36),       -- BSTDX DS3 (CS) redundant IOA
                bIWU-OC3-mm-n-1 (37),   -- BSTDX OC3 (IWU) multimode non
                                        redundant IOA
                bIWU-OC3-smfir-n-1 (38), -- BSTDX OC3 (IWU) single mode
                                        intermediate reach non-redundant
                                        IOA
                toc3-stm1copper-n-4 (39), -- Topaz OC3/STM1 Copper, 4-port
                                        SC, non-redundant
                toc3-stm1copper-r-4 (40), -- Topaz OC3/STM1 Copper, 4-port
                                        SC, redundant
                spa-universal (32),     -- Topaz Universal SPA (E1/T1)
                nplus1-chassis (41),     -- N+1 Chassis Indicator
                toc12-smflr-n-1(42),    -- Topaz OC12/STM4 SMF long
                                        reach non-
                                        redundant IOA
                npa-universal (43),     -- Garnet Universal NPA
                bIWU-OC3-bumm-1 (44),   -- BSTDX OC3 (IWU) base unit
                                        multimode redundant IOA
                bIWU-OC3-bumm-smfir-1 (45), -- BSTDX OC3 (IWU) base unit
                                        single mode intermediate reach
                                        redundant IOA
                bIWU-OC3-trm-mm-1 (46), -- BSTDX OC3 (IWU) tranceiver
                                        module multimode redundant IOA
                bIWU-OC3-trm-smfir-1 (47) -- BSTDX OC3 (IWU) tranceiver
                                        module single mode intermediate
                                        reach redundant IOA
        }
```

```
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The type of IOA attached to this card viewed from NMS."
        ::= { cardEntry 127 }
```

Discard threshold table

```
cardNrtsDiscardTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF CardNrtsDiscardEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A list of ATM NRTS discard threshold configuration
                records."
        ::= { card 3 }
cardNrtsDiscardEntry OBJECT-TYPE
        SYNTAX  CardNrtsDiscardEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A discard threshold configuration record for ATM NRTS."
        INDEX { cardNrtsDiscardLogSlot, cardNrtsDiscardRedundState,
                cardNrtsDiscardIndex }
        ::= { cardNrtsDiscardTable 1 }

CardNrtsDiscardEntry ::=
        SEQUENCE {
                cardNrtsDiscardLogSlot
                    INTEGER,
                cardNrtsDiscardRedundState
                    INTEGER,
                cardNrtsDiscardIndex
                    INTEGER,
                cardNrtsDiscardThresh
                    INTEGER
                }

cardNrtsDiscardLogSlot OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The physical slot ID for this card."
        ::= { cardNrtsDiscardEntry 1 }

cardNrtsDiscardRedundState OBJECT-TYPE
        SYNTAX  INTEGER {
                active (1),
                standby (2)
        }
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The current redundancy state of this card."
        ::= { cardNrtsDiscardEntry 2 }

cardNrtsDiscardIndex OBJECT-TYPE
        SYNTAX INTEGER (1..256)
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The CLP=1 threshold index corresponding to the MCR class."
        ::= { cardNrtsDiscardEntry 3 }

cardNrtsDiscardThresh OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The CLP=1 discard threshold corresponding to the MCR
                class."
        ::= { cardNrtsDiscardEntry 4 }
```

EFCI threshold table

```
cardNrtsEfciTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF CardNrtsEfciEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A list of ATM NRTS EFCI threshold configuration records."
        ::= { card 4 }

cardNrtsEfciEntry OBJECT-TYPE
        SYNTAX  CardNrtsEfciEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "An EFCI configuration record for ATM NRTS."
        INDEX { cardNrtsEfciLogSlot, cardNrtsEfciRedundState,
                cardNrtsEfciIndex }
        ::= { cardNrtsEfciTable 1 }

CardNrtsEfciEntry ::=
        SEQUENCE {
                cardNrtsEfciLogSlot
                    INTEGER,
                cardNrtsEfciRedundState
                    INTEGER,
                cardNrtsEfciIndex
                    INTEGER,
                cardNrtsEfciThresh
                    INTEGER
                }
```

```
cardNrtsEfciLogSlot OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The physical slot ID for this card."
        ::= { cardNrtsEfciEntry 1

cardNrtsEfciRedundState OBJECT-TYPE
        SYNTAX  INTEGER {
                active (1),
                standby (2)
                }
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The current redundancy state of this card."
        ::= { cardNrtsEfciEntry 2 }

cardNrtsEfciIndex OBJECT-TYPE
        SYNTAX INTEGER (1..256)
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The EFCI threshold index corresponding to the MCR class."
        ::= { cardNrtsEfciEntry 3 }

cardNrtsEfciThresh OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The EFCI threshold corresponding to the MCR class.
                Must be less than the CLP=1 threshold for this MCR class."
        ::= { cardNrtsEfciEntry 4 }
```

Rate Increase Factor (RIF) table

```
cardNrtsRifTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF CardNrtsRifEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A list of ATM NRTS RIF configuration records."
        ::= { card 5 }

cardNrtsRifEntry OBJECT-TYPE
        SYNTAX  CardNrtsRifEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A RIF configuration record for ATM NRTS."
        INDEX { cardNrtsRifLogSlot, cardNrtsRifRedundState,
                cardNrtsRifIndex }
        ::= { cardNrtsRifTable 1 }

CardNrtsRifEntry ::=
        SEQUENCE {
                cardNrtsRifLogSlot
                    INTEGER,
                cardNrtsRifRedundState
                    INTEGER,
                cardNrtsRifIndex
                    INTEGER,
                cardNrtsRifValue
                    INTEGER
                }

cardNrtsRifLogSlot OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The physical slot ID for this card."
        ::= { cardNrtsRifEntry 1 }

cardNrtsRifRedundState OBJECT-TYPE
        SYNTAX  INTEGER {
                active (1),
                standby (2)
        }
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The current redundancy state of this card."
        ::= { cardNrtsRifEntry 2 }

cardNrtsRifIndex OBJECT-TYPE
        SYNTAX INTEGER (1..256)
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The Rate Increase Factor index corresponding to the MCR
                class."
        ::= { cardNrtsRifEntry 3 }
```

```
cardNrtsRifValue OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The Rate Increase Factor corresponding to the MCR class.
                Must be less than 16."
        ::= { cardNrtsRifEntry 4 }
```

Rate Decrease Factor (RDF) table

```
cardNrtsRdfTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF CardNrtsRdfEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A list of ATM NRTS RDF configuration records."
        ::= { card 6 }

cardNrtsRdfEntry OBJECT-TYPE
        SYNTAX  CardNrtsRdfEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A RDF configuration record for ATM NRTS."
        INDEX { cardNrtsRdfLogSlot, cardNrtsRdfRedundState,
                cardNrtsRdfIndex }
        ::= { cardNrtsRdfTable 1 }

CardNrtsRdfEntry ::=
        SEQUENCE {
                cardNrtsRdfLogSlot
                    INTEGER,
                cardNrtsRdfRedundState
                    INTEGER,
                cardNrtsRdfIndex
                    INTEGER,
                cardNrtsRdfValue
                    INTEGER
                }

cardNrtsRdfLogSlot OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The physical slot ID for this card."
        ::= { cardNrtsRdfEntry 1 }

cardNrtsRdfRedundState OBJECT-TYPE
        SYNTAX  INTEGER {
                active (1),
                standby (2)
        }
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The current redundancy state of this card."
        ::= { cardNrtsRdfEntry 2 }

cardNrtsRdfIndex OBJECT-TYPE
        SYNTAX INTEGER (1..256)
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The Rate Decrease Factor index corresponding to the MCR
                class."
        ::= { cardNrtsRdfEntry 3 }

cardNrtsRdfValue OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The Rate Decrease Factor corresponding to the MCR class.
                Must be less than 16."
        ::= { cardNrtsRdfEntry 4 }

cardNtpRefTable  OBJECT-TYPE
        SYNTAX SEQUENCE OF CardNtpRefEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "A list of cardNtpRefEntry's."
        ::= { card 7}

cardNtpRefEntry  OBJECT-TYPE
        SYNTAX CardNtpRefEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "The card ntp reference entry contains the configuration
                and status information pertaining to the time servers
                referenced by the cards."
         INDEX { cardNtpLogicalSlotId, cardNtpRedundState,
                cardNtpPeerIndex }
        ::= { cardNtpRefTable 1}
```

```
CardNtpRefEntry  ::=
        SEQUENCE {
                cardNtpLogicalSlotId
                    INTEGER,
                cardNtpRedundState
                    INTEGER,
                cardNtpPeerIndex
                    INTEGER,
                cardNtpPeerAddr
                    IpAddress,
                cardNtpPeerStatus
                    INTEGER,
                cardNtpReachableStatus
                    INTEGER,
                cardNtpOrgTimestampISec
                    INTEGER,
                cardNtpOrgTimeStampFSec
                    INTEGER,
                cardNtpOffset
                    INTEGER,
                cardNtpMaxOffset
                    INTEGER,
                cardNtpNumberofPolls
                    INTEGER,
                cardNtpNumofFailedPolls
                    INTEGER,
                cardNtpReset
                    INTEGER
                }

cardNtpLogicalSlotId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Card Logical Slot"
        ::= { cardNtpRefEntry 1 }

cardNtpRedundState OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Card Redundancy State"
        ::= { cardNtpRefEntry 2 }

cardNtpPeerIndex OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Ntp Peer Table Index"
        ::= { cardNtpRefEntry 3 }

cardNtpPeerAddr  OBJECT-TYPE
        SYNTAX IpAddress
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This is the IP Address the NTP peer reference server"
        ::= { cardNtpRefEntry 4 }

cardNtpPeerStatus OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "clock selection status
                0 - rejected
                1 - Passed sanity Checks"
        ::= { cardNtpRefEntry 5 }

cardNtpReachableStatus OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Reachable status
                0 - not reachable
                1 - reachable"
        ::= { cardNtpRefEntry 6 }

cardNtpOrgTimestampISec OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS  mandatory
        DESCRIPTION
                "Seconds portion of last time the local host sent a
                time request message to the peer"
        ::= { cardNtpRefEntry 7 }

cardNtpOrgTimeStampFSec OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS  mandatory
        DESCRIPTION
                "Fractions of a second part of the last time the
                local host sent a time request message to the peer"
        ::= { cardNtpRefEntry 8 }

cardNtpOffset OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS  mandatory
        DESCRIPTION
                "Deviation between the local ntp clock and the
                last message received from the peer in msec"
        ::= { cardNtpRefEntry 9}
```

```
cardNtpMaxOffset OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS  mandatory
        DESCRIPTION
                "Maximum Deviation between the local ntp clock and the
                peer clock in msec. This value
                shall be cleared after it is read."
        ::= { cardNtpRefEntry 10}

cardNtpNumberofPolls OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS  mandatory
        DESCRIPTION
                "Number of times the local ntp client polled the peer
                reference server for time."
        ::= { cardNtpRefEntry 11}

cardNtpNumofFailedPolls OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS  mandatory
        DESCRIPTION
                "Number of times the peer didn't respond to the
                local ntp client polls."
        ::= { cardNtpRefEntry 12}

cardNtpReset OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS  mandatory
        DESCRIPTION
                "A value of one restarts the ntp process"
        ::= { cardNtpRefEntry 13}
```

The Physical Port Group

```
--  The variables that configure physical ports at a node

pportNumber OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of physical ports (regardless of their current
                state) present at this node."
        ::= { pport 1 }

pportTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF PportEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A list of physical port entries.  The number of entries is
                given by the value of pportNumber."
        ::= { pport 2 }

pportEntry OBJECT-TYPE
        SYNTAX  PportEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "The physical port entry contains objects relevant to a
                physical port."
        INDEX   { pportSlotId,
                  pportId }
        ::= { pportTable 1 }

PportEntry ::=
        SEQUENCE {
                pportSlotId
                    INTEGER,
                pportId
                    INTEGER,
                pportAdminType
                    INTEGER,
                pportNumLport
                    INTEGER,
                pportDataRate
                    INTEGER,
                pportType
                    INTEGER,
                pportRecvClock
                    INTEGER,
                pportXmitClock
                    INTEGER,
                pportAdminStatus
                    INTEGER,
                pportOperStatus
                    INTEGER,
                pportDs1LineType
                    INTEGER,
                pportDs1ZeroCoding
                    INTEGER,
                pportDs1LineBuildout
                    INTEGER,
                pportDiagTestId
                    INTEGER,
                pportDiagTestRuns
                    INTEGER,
                pportInOctets
                    Counter,
                pportInFrames
```

```
    Counter,
pportInDiscards
    Counter,
pportInErrors
    Counter,
pportOutOctets
    Counter,
pportOutFrames
    Counter,
pportOutDiscards
    Counter,
pportOutErrors
    Counter,
pportDiagState
    INTEGER,
pportDiagOptionStr
    OCTET STRING,
pportDiagPassCount
    INTEGER,
pportDiagFailCount
    INTEGER,
pportDiagResultStr
    DisplayString,
pportLinkDownReason
    INTEGER,
pportInterface
    INTEGER,
pportAdminInterface
    INTEGER,
pportCellScramble
    INTEGER,
pportCbitParity
    INTEGER,
pportMaxBufferSize
    INTEGER,
pportPeakCellRate0
    INTEGER,
pportPeakCellRate1
    INTEGER,
pportPeakCellRate2
    INTEGER,
pportPeakCellRate3
    INTEGER,
pportPeakCellRate4
    INTEGER,
pportPeakCellRate5
    INTEGER,
pportPeakCellRate6
    INTEGER,
pportPeakCellRate7
    INTEGER,
pportInCells
    Counter,
pportInErrorCells
    Counter,
pportOutCells
    Counter,
pportDs3LineBuildout
    INTEGER,
pportSetDS0LoopUp
    INTEGER,
pportSetDS0LoopDown
    INTEGER,
pportDS0LoopUpStatus
    INTEGER,
pportDS0LoopDownStatus
    INTEGER,
pportDS0LoopStatus
    INTEGER,
pportISDN
    INTEGER,
pportdsx3LoopbackConfig
    INTEGER,
pportdsx3SendCode
    INTEGER,
pportdsx3LoopStatus
    INTEGER,
pportdsx3FEACStatus
    INTEGER,
pportds1LoopbackConfig
    INTEGER,
pportds1SendCode
    INTEGER,
pportds1LoopStatus
    INTEGER,
pportSetClkBkup
    INTEGER,
pportAtmIdleWord
    INTEGER,
pportAtmDisCardMode
    INTEGER,
pportAtmLastUnconfiguredVpi
    INTEGER,
pportAtmLastUnconfiguredVci
    INTEGER,
pportAtmUnconfiguredCells
    Counter,
pportAtmNumBitsVCI
    INTEGER,
pportAtmNumBitsVPI
    INTEGER,
pportAtmInterfaceType
    INTEGER,
pportSonetSDHLoopbackConfig
    INTEGER,
pportSonetSDHLoopStatus
    INTEGER,
pportOutDiscardsCell
```

```
    Counter,
pportAtmPlcp
    INTEGER,
pportCbrTargetClockMode
    INTEGER,
pportCbrCurrentClockMode
    INTEGER,
-- 75 is deprecated
-- This OID will be reused.
pportFiberType
    INTEGER,
pportLaserStatus
    INTEGER,
pportMaxActiveVpiBits
    INTEGER,
pportBipErrorsThresh
    INTEGER,
pportBipSectionErrors
    Counter,
pportBipLineErrors
    Counter,
pportBipPathErrors
    Counter,
pportFebeErrors
    Counter,
pportHcsErrors
    Counter,
pportHcsSevereErrors
    Counter,
pportCongestedReceivedCells
    Counter,
pportCongestedTransmittedCells
    Counter,
pportAtmLayerErroredReceivedCells
    Counter,
pportAtmLayerErroredTransmittedCells
    Counter,
pportDS0BitStuff
    INTEGER,
pportDS0BitErrorCount
    INTEGER,
pportDS0BitErrorFreeSeconds
    INTEGER,
pportDS0BitErroredSeconds
    INTEGER,
pportDS0MidspanRepeaters
    INTEGER,
pportDS0TestPatternSync
    INTEGER,
pportDS0InjectBitError
    INTEGER,
pportDS0FarendLpbkType
    INTEGER,
pportDS0LpbkMode
    INTEGER,
pportDS0SwitchLpbkStart
    INTEGER,
pportDS0SwitchLpbkEnd
    INTEGER,
pportDS0FarendDS0InLpbk
    INTEGER,
pportDS0SendTestTraffic
    INTEGER,
pportOc3LoopConfig
    INTEGER,
pportOc3LoopStatus
    INTEGER,
pportISDNIpBaseAddr
    IpAddress,
pportSonetSTM1Scramble
    INTEGER,
pportEFCIMarking
    INTEGER,
pportAtmQOSTransmitMode
    INTEGER,
pportHECMode
    INTEGER,
pportISDNChannelStatus
    OCTET STRING,
 pportds1FarEndLoopStatus
    INTEGER,
 pportds1FDLControl
    INTEGER,
 pportds1FDLPrmXmit
    INTEGER,
 pportds1FDLPidXmit
    INTEGER,
 pportds1FDLXmitPid
    OCTET STRING,
 pportds1FDLRcvPid
    OCTET STRING,
 pportds1FDLRcvTsid
    OCTET STRING,
pportSonetSDHFramingMode
    INTEGER,
 pportds1InbandLoopType
    INTEGER,
 pportESFDataLinkStatus
    INTEGER,
 pportPMTcaId
    INTEGER,
 pportBchanTimerValue
        INTEGER,
pportAAL5CRC32Error
    Counter,
pportAAL5CPIError
    Counter,
pportAAL5LengthError
```

```
                Counter,
            pportAAL5ReassemblyTimerError
                Counter,
            pportAAL5MaxNrSegError
                Counter,
             pportRedundancyArch
                INTEGER,
             pportAPSadminDir
                    INTEGER,
             pportAPSlineType
                    INTEGER,
             pportAPSrevertiveMode
                    INTEGER,
             pportAPSpairedSlotId
                    INTEGER,
             pportAPSpairedPportId
                    INTEGER,
             pportAPSsfBerThresh
                    INTEGER,
             pportAPSsdBerThresh
                    INTEGER,
             pportAPSwtrPeriod
                    INTEGER,
             pportAPSprotectionLineState
                    INTEGER,
             pportAPSxCommand
                INTEGER,
             pportAPSconfigStatus
                    INTEGER,
             pportAPSOperRxStatus
                    INTEGER
            }

pportSlotId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The slot number of the corresponding physical port."
        ::= { pportEntry 1 }

pportId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The port number of the corresponding physical port on the
                board."
        ::= { pportEntry 2 }
```

```
pportAdminType OBJECT-TYPE
        SYNTAX INTEGER {
                v35-6 (1),              -- 6-port V.35 i/o card
                ft1-1-24 (2),           -- 1-port 24-channel Fractional
                                        T1
                fe1-1-30 (3),           -- 1-port 30-channel Fractional
                                        E1
                uio-6 (4),              -- 6-port universal i/o card
                cp1 (5),                -- Control Processor
                uio-8 (6),              -- 8-port V.35 i/o card
                ft1-4-24 (7),           -- 4-port 24-channel Fractional
                                        T1
                fe1-4-30 (8),           -- 4-port 30-channel Fractional
                                        E1
                ft3-1 (9),              -- 1-port Fractional T3 i/o card
                fe3-1 (10),             -- 1-port Fractional E3 i/o card
                hssi-2 (11),            -- HSSI i/o card
                dsx1-10 (12),           -- 10-port DSX-1 card
                rs232-18 (13),          -- 18-port X.21/V.24 I/O card,
                                        for STDX
                                        3000/6000 only
                rs232-8 (14),           -- 8-port X.21/V.24 I/O card,
                                        for STDX 3000/
                                        6000 only
                ut1-4-24 (15),          -- 4-port 24-channel Un-
                                        Channelized T1
                ue1-4-30 (16),          -- 4-port 30-channel Un-
                                        Channelized E1
                atmds3-1 (17),          -- 1-port ATM DS3 UNI I/O card
                atme3-1 (18),           -- 1-port ATM E3 UNI I/O card
                pri-4 (19),             -- 4-port ISDN PRI I/O card
                e1-pri-4 (20),          -- 4-port E1 PRI I/O card
                sft1-4-24 (21),         -- 4-port short haul 24-channel
                                        Fractional T1 card
                sut1-4-24 (22),         -- 4-port short haul 24-channel
                                        Un-Channelized T1 card
                st1-pri-4 (23),         -- 4-port short haul PRI I/O card
                t1-atm   (24),          -- T1 ATM
                e1-atm   (25),          -- E1 ATM
                ads3-t3   (26),         -- ATM DS3 T3 (topaz)
                ads3-e3   (27),         -- ATM DS3 E3 (topaz)
                cbr-ds1-s-4 (28),       -- 4-port CBR T1 structured card
                cbr-ds1-us-4 (29),      -- 4-port CE T1 card
                cbr-e1-s-4 (30),        -- 4-port CBR E1 structured card
                cbr-e1-us-4 (31),       -- 4-port CE E1 card
                cbr-atmiwu-1 (32),      -- 1-port ATM-IWU STM-1/STS-3c
                                        card
                toc3-atm-4 (33),        -- 4-port Topaz OC3c ATM card
                tstm1-atm-4 (34),       -- 4-port Topaz STM1 ATM card
                atmcs-1     (37),       -- 1-port ATM-CS card (siemens)
                toc12-atm-1 (38),       -- 1-port Topaz OC12c ATM card
                tstm4-atm-1 (39),       -- 1-port Topaz STM4 ATM card
                ads1-t1-8   (40),       -- 8 port Topaz T1 ATM T1 card
                ads1-e1-8   (41),       -- 8 port Topaz T1 ATM E1 card
```

```
                ads1-j2-8   (42),         -- 8 port Topaz T1 ATM J2 card
                e1-12 (43),               -- 12-port E1 i/o card
                bio1_4_16 (44),           -- Garnet BIO1 4 PHY sub-cards 16
                                          ports
                bio1_oc3_4 (45),          -- Garnet BIO1 OC3 PHY sub-card 4
                                          ports
                bio1_oc12_1 (46),         -- Garnet BIO1 OC12 PHY sub-card
                                          1 port
                bio1_oc12x4 (47),         -- Garnet BIO1 OC12x4 PHY sub-
                                          card 1
                port 4 channels
                bio1_oc48_1 (48),         -- Garnet BIO1 OC48 PHY sub-card
                                          1 port
                np1 (49),                 -- Garnet Node Processor card
                sf1 (50),                 -- Garnet Switch Fabric card
                tm1 (51),                 -- Garnet Timing Module card
                tfds3-t3-6    (52),       -- 6-port Topaz DS3 T3 Ultracore
                                          card
                tfds3-e3-6    (53),       -- 6-port Topaz DS3 E3 Ultracore
                                          card
                tfast-ether-4 (54),       -- 4-port Topaz Fast Ethernet
                                          Ultracore card
                fast-ether-2  (55),       -- 2-port BSTDX Fast Ethernet
                                          Ultracore card
                ls-oc3-1     (56)         -- 1-port BSTDX OC3c/STM-1
                                          Ultracore card
                }
                ACCESS read-write
                STATUS mandatory
                DESCRIPTION
                        "The defined type of the board which the
                        physical port is on."
                ::= { pportEntry 3 }

pportNumLport OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of logical ports on the physical port."
        ::= { pportEntry 4 }

pportDataRate OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "An estimate of the physical port's data rate in bits
                per second. The data rate can't be an arbitrary value. The
                legal values are list below. Note that the T1/E1 class of
                cards don't support this variable and return a 0

        For UIO, DSX1:
                9600, 19200, 38400, 48000, 56000, 64000,
                128000, 192000, 256000, 320000, 384000, 448000,
                512000, 576000, 640000, 704000, 768000, 832000,
                896000, 960000, 1024000, 1088000, 1152000, 1216000,
                1280000, 1344000, 1408000, 1472000, 1536000, 1600000,
                1664000, 1728000, 1792000, 1856000, 1920000, 1984000,
                2048000, 4096000, 6144000, 8192000
        For HSSI:
                1579000, 3158000, 4737000, 6316000, 7895000, 9474000,
                11053000, 12632000, 14211000, 15790000, 17369000,
                18948000, 20527000, 22106000, 23685000, 25264000,
                26843000, 28422000, 30001000, 31580000, 33159000,
                34738000, 36317000, 37896000, 39475000, 41054000,
                42633000, 44212000, 45791000, 47370000, 48949000,
                50528000
        For ATM DS3, CHANNELIZED DS3:
                44736000

        For ATM-IWU, TOPAZ OC3c:
                155520000

        For T1, E1, UT1, UE1, PRI, SH-T1:
                0
          "
        ::= { pportEntry 5 }

pportType OBJECT-TYPE
        SYNTAX INTEGER {
                v35-6 (1),      -- 6-port V.35 i/o card
                ft1-1-24 (2),   -- 1-port 24-channel Fractional T1
                fe1-1-30 (3),   -- 1-port 30-channel Fractional E1
                uio-6 (4),      -- 6-port universal i/o card
                cp1 (5),        -- Control Processor
                uio-8 (6),      -- 8-port V.35 i/o card
                ft1-4-24 (7),   -- 4-port 24-channel Fractional T1
                fe1-4-30 (8),   -- 4-port 30-channel Fractional E1
                ft3-1 (9),      -- 1-port Fractional T3 i/o card
                fe3-1 (10),     -- 1-port Fractional E3 i/o card
                hssi-2 (11),    -- HSSI i/o card
                dsx1-10 (12),   -- 10-port DSX-1 card
                rs232-18 (13),   -- 18-port X.21/V.24 I/O module, for STDX
                                3000/6000 only
                rs232-8 (14),   -- 8-port X.21/V.24 I/O card, for STDX 3000/
                                6000 only
                ut1-4-24 (15),   -- 4-port 24-channel Un-Channelized T1
                ue1-4-30 (16),   -- 4-port 30-channel Un-Channelized E1
                atmds3-1 (17),   -- 1-port ATM DS3 UNI I/O card
                atme3-1 (18),   -- 1-port ATM E3 UNI I/O card
                pri-4 (19),     -- 4-port ISDN PRI I/O card
                e1-pri-4 (20),   -- 4-port E1 PRI I/O card
                sft1-4-24 (21),-- 4-port short haul 24-channel
                                Fractional T1 card
                sut1-4-24 (22),-- 4-port short haul 24-channel Un-
                                Channelized T1 card
                st1-pri-4 (23),           -- 4-port short haul PRI I/O card
```

```
            t1-atm   (24),          -- T1 ATM
            e1-atm   (25),          -- E1 ATM
            ads3-t3   (26),         -- ATM DS3 T3 (topaz)
            ads3-e3   (27),         -- ATM DS3 E3 (topaz)
            cbr-ds1-s-4 (28),       -- 4-port CBR T1 structured card
            cbr-ds1-us-4 (29),      -- 4-port CE T1 card
            cbr-e1-s-4 (30),        -- 4-port CBR E1 structured card
            cbr-e1-us-4 (31),       -- 4-port CE E1 card
            cbr-atmiwu-1 (32),      -- 1-port ATM-IWU STM-1/STS-3c
                                    card
            toc3-atm-4 (33),        -- 4-port Topaz OC3c ATM card
            tstm1-atm-4 (34),       -- 4-port Topaz STM1 ATM card
            atmcs-1     (37),       -- 1-port ATM-CS card (siemens)
            toc12-atm-1 (38),       -- 1-port Topaz OC12c ATM card
            tstm4-atm-1 (39),       -- 1-port Topaz STM4 ATM card
            ads1-t1-8   (40),       -- 8 port Topaz T1 ATM T1 card
            ads1-e1-8   (41),       -- 8 port Topaz T1 ATM E1 card
            ads1-j2-8   (42),       -- 8 port Topaz T1 ATM J2 card
            e1-12 (43),             -- 12-port E1 i/o card
            bio1_4_16 (44),         -- Garnet BIO1 4 PHY sub-cards 16
                                    ports
            bio1_oc3_4 (45),        -- Garnet BIO1 OC3 PHY sub-card 4
                                    ports
            bio1_oc12_1 (46),       -- Garnet BIO1 OC12 PHY sub-card
                                    1 port
            bio1_oc12x4 (47),       -- Garnet BIO1 OC12x4 PHY sub-
                                    card 1 port 4 channels
            bio1_oc48_1 (48),       -- Garnet BIO1 OC48 PHY sub-card
                                    1 port
            np1 (49),               -- Garnet Node Processor card
            sf1 (50),               -- Garnet Switch Fabric card
            tm1 (51),               -- Garnet Timing Module card
            tfds3-t3-6    (52),     -- 6-port Topaz DS3 T3 Ultracore
                                    card
            tfds3-e3-6    (53),     -- 6-port Topaz DS3 E3 Ultracore
                                    card
            tfast-ether-4 (54),     -- 4-port Topaz Fast Ethernet
                                    Ultracore card
            fast-ether-2  (55),     -- 2-port BSTDX Fast Ethernet
                                    Ultracore card
            ls-oc3-1     (56)       -- 1-port BSTDX OC3c/STM-1
                                    Ultracore card
    }
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The actual type of the physical port."
        ::= { pportEntry 6 }
```

```
pportRecvClock OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The receive clock source. This object works in
                cooperation with pportXmitClock.

                The following are the various values for different clock
                source types:
                Type  Clock Source   pportRevClock      pportXmitClock
                    { pportEntry 7 }   { pportEntry 8 }
                V.35  DCE ............ 0                  0
                or UIO  LoopTime DCE ... 3                  0
                DTE .............        1                  2
                Direct Trunk ...          1                  0
                Others  Loop Timed ..... 1                  1
                Internal .......          2                  2
                External .......    don't care          3
                Chassis  .......          1                  4
                "
        ::= { pportEntry 7 }

pportXmitClock OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The transmit clock source. This object works in
                cooperation with pportXmitClock. The following are the
                various values for different clock
                source types:
                Type  Clock SourcepportRevClockpportXmitClock
                                { pportEntry 7 }{ pportEntry 8 }
                V.35  DCE ............0        0
                or UIO  LoopTime DCE .3        0
                DTE .............      1        2
                Direct Trunk ...       1        0
                Others  Loop Timed ...1        1
                Internal .......       2        2
                External ....... don't care    3
                Chassis  .......      1       4
                "
        ::= { pportEntry 8 }
```

```
pportAdminStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                    invalid (0),
                    up (1),
                    down (2),
                    testing (3)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The desired state of the physical port."
        ::= { pportEntry 9 }

pportOperStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                    up (1),
                    down (2),
                    testing (3)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The current state of the physical port."
        ::= { pportEntry 10 }

pportDs1LineType OBJECT-TYPE
        SYNTAX INTEGER {
                d4 (1),
                esf-ansi (2),
                esf-att-address-a (3),
                esf-none (4),
                e1-cas-crc4 (5),
                e1-cas-no-crc4 (6),
                e1-no-cas-crc4 (7),
                e1-no-cas-no-crc4 (8),
                esf-att-address-b (9),
                sf-ansi (10),
                e1-unstructured (11)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The line type of the T1 or FT1-24B port. ANSI ESF is
                equivalent to Bellcore ESF."
        ::= { pportEntry 11 }

pportDs1ZeroCoding OBJECT-TYPE
        SYNTAX INTEGER {
                ami (1),
                b8zs (2),
                hdb3 (3),
                jammed-bit (4),
                ami-nostuff (5)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The variety of zero code suppression used on the T1
                or FT1-24B link.  Jammed bit is equivalent to dsx1JBZS
                where the DS0s will run at 56K bps."
        ::= { pportEntry 12 }

pportDs1LineBuildout OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The line build out or line length specification for
                the DS1 or DSX-1 transmitter. These values are card
                specific:

                10 port DSX-1 card
                -----------------
                The accepted values are:

                1  -   0 to 110 feet
                2  - 110 to 220 feet
                3  - 220 to 330 feet
                4  - 330 to 440 feet
                5  - 440 to 550 feet
                6  - 550 to 660 feet
                7  - over 655 feet

                1-port and 4-port T1 cards
                --------------------------
                The accepted values are:

                96  -  0 to 133 feet
                128  - 133 to 266 feet
                160  - 266 to 399 feet
                192  - 399 to 533 feet
                224  - 533 to 655 feet
                96   - 0 db
                32   - -7.5 db
                64   - -15 db

                4-port circuit emulation short haul T1 card
                -------------------------------------------
                The accepted values are:

                96  -  0 to 133 feet (0.6 dB cable loss)
                128  - 133 to 266 feet (1.2 dB cable loss)
                160  - 266 to 399 feet (1.8 dB cable loss)
                192  - 399 to 533 feet (2.4 dB cable loss)
                224  - 533 to 655 feet (3.0 dB cable loss)

                4-port Short Haul T1 cards
                --------------------------
```

```
                The accepted values are:
                96  -  0 to 133 feet
                128  - 133 to 266 feet
                160  - 266 to 399 feet
                192  - 399 to 533 feet
                224  - 533 to 655 feet


                8 Port Topaz T1 ATM Card
                ------------------------
                The accepted values are:

                1  -   0 to 133 feet
                2  - 133 to 266 feet
                3  - 266 to 399 feet
                4  - 399 to 533 feet
                5  - 533 to 655 feet
                 6  - E1 Short Haul
                 7  - E1 Long Haul"

        ::= { pportEntry 13 }

pportDiagTestId OBJECT-TYPE
        SYNTAX  INTEGER {
                    v35-sca-local-loopback (1),
                    v35-sca-remote-loopback (2),
                    v35-csu-loopback (3),
                    hssi-local-dte-loopback (5),
                    hssi-local-line (6),
                    hssi-remote-line-loopback (7),
                    ds1-framer-local-loopback (11),
                    ds1-line-local-loopback (12),
                    ds1-framer-remote-loopback (13),
                    ds1-line-remote-loopback (14)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Identification for the diagnostics tests to be run."
        ::= { pportEntry 14 }

pportDiagTestRuns OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The number of passes of the diagnostics tests to be run.
                The default value is 1."
        ::= { pportEntry 15 }

pportInOctets OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of octets received on the physical port,
                including framing characters."
        ::= { pportEntry 16 }

pportInFrames OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of frames received on the physical port."
        ::= { pportEntry 17 }

pportInDiscards OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of inbound frames discarded."
        ::= { pportEntry 18 }
```

```
pportInErrors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of inbound frames that contained
                erroneous headers (e.g., illegal or unknown DLCIs)."
        ::= { pportEntry 19 }

pportOutOctets OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of octets transmitted out of the physical
                port, including framing characters."
        ::= { pportEntry 20 }

pportOutFrames OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of frames transmitted out of the
                physical port."
        ::= { pportEntry 21 }

pportOutDiscards OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of outbound frames discarded due to
                severe congestion."
        ::= { pportEntry 22 }

pportOutErrors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of outbound frames that causes xmit
                errors."
        ::= { pportEntry 23 }

pportDiagState OBJECT-TYPE
        SYNTAX  INTEGER {
                inactive (0),
                active(1)
        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The current state of the diagnostic on this physical
                port."
        ::= { pportEntry 24 }

pportDiagOptionStr OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Optional parameters to the diagnostic."
        ::= { pportEntry 25 }

pportDiagPassCount OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Number of successful diagnostic passes."
        ::= { pportEntry 26 }

pportDiagFailCount OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Number of failed diagnostic passes."
        ::= { pportEntry 27 }

pportDiagResultStr OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Description of last diagnostic failure."
        ::= { pportEntry 28 }
```

```
pportLinkDownReason OBJECT-TYPE
        SYNTAX  INTEGER {
                none                    (0),
                red-alarm               (1),
                yellow-alarm            (2),
                blue-alarm              (4),
                carrier-loss            (8),
                looped-back             16),
                ber-threshold           (64),
                signal-label-mismatch   (128),
                loss-of-signal          (256),
                loss-of-frame           (512),
                loss-of-cell-delination (1024),
                line-AIS                (2048),
                path-AIS                (4096),
                loss-of-pointer         (8192),
                line-RFI                (16384),
                path-RFI                (32768),
                signal-label-undefined  (65536)
                idle                    (131072),
                 equipment-mismatch     (262144),
                 admin-down             (524288)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Reason why the link is down.  The blue-alarm is equivalent
                to the Alarm Indication Signal (AIS) failure."
        ::= { pportEntry 29 }

pportInterface OBJECT-TYPE
        SYNTAX  INTEGER {
                eia449          (1),
                x21             (2),
                eia530          (3),
                eia530A         (4),
                v35             (5),
                e1-coax         (6),
                e1-db           (7),
                none            (8),
                v24             (9),
                sonet           (10),
                sdh             (11),
                multi-mode      (12),       -- optical interface for topaz
                                               only
                single-mode     (13)        -- optical interface for topaz
                                               only
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Actual interface connected to Multi-Interface Port or ATM-
                IWU."
        ::= { pportEntry 30 }

pportAdminInterface OBJECT-TYPE
        SYNTAX INTEGER {
                eia449  (1),
                x21  (2),
                eia530  (3),
                eia530A  (4),
                v35  (5),
                e1-coax  (6),
                e1-db  (7),
                none  (8),
                v24  (9),
                sonet  (10),
                sdh  (11),
                multi-mode (12), -- optical interface for topaz only
                single-mode (13), -- optical interface for topaz only
                v35-nrzi (14)    -- only for SDLC FRAD on UIO
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Configured interface connected to Multi-Interface Port or
                ATM-IWU."
        ::= { pportEntry 31 }

pportCellScramble OBJECT-TYPE
        SYNTAX  INTEGER {
                disabled (1),
                enabled(2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Configured ATM cell payload scrambling."
        ::= { pportEntry 32 }

pportCbitParity OBJECT-TYPE
        SYNTAX  INTEGER {
                disabled (1),
                enabled (2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Configured ATM DS3 C-bit Parity."
        ::= { pportEntry 33 }

pportMaxBufferSize OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Maximum receive/transmit buffer size for the ATM UNI DS3
                IOP"
        ::= { pportEntry 34 }
```

```
pportPeakCellRate0 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Peak Cell Rate (cell/sec.) for rate queue #0.
                 rate queues 0 - 3 are high priority"
        ::= { pportEntry 35 }

pportPeakCellRate1 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Peak Cell Rate (cell/sec.) for rate queue #1.
                 rate queues 0 - 3 are high priority"
        ::= { pportEntry 36 }

pportPeakCellRate2 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Peak Cell Rate (cell/sec.) for rate queue #2.
                 rate queues 0 - 3 are high priority"
        ::= { pportEntry 37 }

pportPeakCellRate3 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Peak Cell Rate (cell/sec.) for rate queue #3.
                 rate queues 0 - 3 are high priority"
        ::= { pportEntry 38 }

pportPeakCellRate4 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Peak Cell Rate (cell/sec.) for rate queue #4.
                 rate queues 4 - 7 are low priority"
        ::= { pportEntry 39 }

pportPeakCellRate5 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Peak Cell Rate (cell/sec.) for rate queue #5.
                 rate queues 4 - 7 are low priority"
        ::= { pportEntry 40 }

pportPeakCellRate6 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Peak Cell Rate (cell/sec.) for rate queue #6.
                 rate queues 4 - 7 are low priority"
        ::= { pportEntry 41 }

pportPeakCellRate7 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Peak Cell Rate (cell/sec.) for rate queue #7.
                 rate queues 4 - 7 are low priority"
        ::= { pportEntry 42 }

pportInCells OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of cells received"
        ::= { pportEntry 43 }

pportInErrorCells OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of cells received with error"
        ::= { pportEntry 44 }

pportOutCells OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of cells transmitted"
        ::= { pportEntry 45 }
```

```
pportDs3LineBuildout OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The line build out or line length specification for
                the DS3 transmitter.

                1-port ATM-UNI DS3 card
                -------------------------
                The accepted values are:

                1 - 0 to 225 feet
                2 - 226 to 450 feet"
        ::= { pportEntry 46 }

pportSetDS0LoopUp OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Bit map used to set one or more DS0's into loopback. Low
                bit corresponds to DS0 1. If a bit is set that DS0 is put
                into loopback."
        ::= { pportEntry 47 }

pportSetDS0LoopDown OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Bit map used to set one or more DS0's out of loopback.
                Low bit corresponds to DS0 1. If a bit is set that DS0 is
                put out of loopback."
        ::= { pportEntry 48 }

pportDS0LoopUpStatus OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Bit map reporting which DS0's have gone into loopback
                since the last time this variable was read"
        ::= { pportEntry 49 }

pportDS0LoopDownStatus OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Bit map reporting which DS0's have gone out of loopback
                since the last time this variable was read"
        ::= { pportEntry 50 }

pportDS0LoopStatus OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Bit map reporting which DS0's are currently in loopback.
                Low bit corresponds to DS0 1"
  ::= { pportEntry 51 }

pportISDN OBJECT-TYPE
        SYNTAX  INTEGER {
                disabled (1),
                enabled  (2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Specifies whether ISDN PRI is enabled for this physical
                port"
        ::= { pportEntry 52 }
```

```
pportdsx3LoopbackConfig OBJECT-TYPE
        SYNTAX INTEGER {
                dsx3NoLoop(1),
                dsx3PayloadLoop (2),
                dsx3LineLoop(3),
                dsx3DiagLoop(4)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Integer used to set the loopback state of the DS3 or E3"
        ::= { pportEntry 53 }

pportdsx3SendCode OBJECT-TYPE
        SYNTAX INTEGER {
                dsx3SendNoCode   (1),
                dsx3SendLineCode  (2),
                dsx3SendResetCode (4),
                dsx3SendUserCode (8)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Integer used to send loopback code to far-end DS3
                User specified code can be set in cascadepm/cascadeds3 mib
                variable dsx3FEACCode in dsx3ConfigTable table."
        ::= { pportEntry 54 }

pportdsx3LoopStatus OBJECT-TYPE
        SYNTAX INTEGER {
                noloop    (1),
                payloadloop (2),
                lineloop    (3),
                diagloop    (4),
                farloopInit (5)
        }
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Integer indicating the current loopback status of the DS3
                or E3"
        ::= { pportEntry 55 }

pportdsx3FEACStatus OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Integer indicating the current DS3 FEAC status:
                none                        (0),
                DS3EquipmentFailure         (1),
                DS3LOS                      (2),
                DS3OutofFrame               (4),
                DS3AISreceived              (8),
                DS3IDLEreceived             (16),
                DS3NonServiceAffectingEquipFailure  (32),
                CommonEquipmentFailure          (64),
                DS3LoopbackReceived         (128),
                DS1ServiceAffectingEquipmentFailure (256),
                DS1NonServiceAffectingEquipFailure  (512),
                SingleDS1LOS                (1024),
                MultipleDS1sLOS             (2048) "
        ::= { pportEntry 56 }

pportds1LoopbackConfig OBJECT-TYPE
        SYNTAX  INTEGER {
                ds1NoLoop      (1),
                ds1PayloadLoop (2),
                ds1LineLoop    (3)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Integer used to set the loopback state of the DS1"
        ::= { pportEntry 57 }
```

```
pportds1SendCode OBJECT-TYPE
        SYNTAX  INTEGER {
                ds1SendNoCode                       (1),
                ds1SendFramedInbandLineActuateLoop  (2),
                ds1SendFramedInbandLineReleaseLoop  (3),
                ds1SendUnframedInbandLineActuateLoop (4),
                ds1SendUnframedInbandLineReleaseLoop (5),
                ds1SendFdlESFAnsiLineActuateLoop    (6),
                ds1SendFdlESFAnsiLineReleaseLoop    (7),
                ds1SendFdlESFAnsiPayloadActuateLoop  (8),
                ds1SendFdlESFAnsiPayloadReleaseLoop  (9)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Integer used to send loopback code to far-end DS1"
        ::= { pportEntry 58 }

pportds1LoopStatus OBJECT-TYPE
                SYNTAX INTEGER {
                                noloop     (1),
                                payloadloop(2),
                                lineloop   (3),
                                diagFramerLoop(4),
                                diagLIULoop(5),
                                diagExternalLoop(6),
                                farEndFramedInbandLineLoop(7),
                                farEndUnframedInbandLineLoop(8),
                                farEndFdlLineLoop(9),
                                farEndFdlPayloadLoop(10)
                                }
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                    "Integer indicating the current loopback status of the
                    DS1"
        ::= { pportEntry 59 }

pportSetClkBkup OBJECT-TYPE
        SYNTAX  INTEGER {
                internalClkBkup     (1),
                looptimedClkBkup    (2)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The desired clock source backup if the card is set in
                external clock source mode"
        ::= { pportEntry 60 }

pportAtmIdleWord  OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The word used to stuff the payload of the ATM idle cell"
        ::= { pportEntry 61 }

pportAtmDisCardMode  OBJECT-TYPE
        SYNTAX  INTEGER {
                ansiInval               (1),
                ansiUnassignedInval     (2),
                atmFInvalid             (3),
                atmFUnassignedInval     (4),
                ccittIdle               (5),
                ccittUnassignedIdle     (6)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The desired clock source backup if the card is set in
                external clock source mode"
        ::= { pportEntry 62 }

pportAtmLastUnconfiguredVpi OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This entry holds the Vpi read from the last Unconfigured
                atm cell received.  This entry applies to ATM pports only."
        ::={ pportEntry 63 }

pportAtmLastUnconfiguredVci OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This entry holds the Vci read from the last Unconfigured
                atm cell received.  This entry applies to ATM pports only."
        ::={ pportEntry 64 }

pportAtmUnconfiguredCells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This entry holds a count of the number of Unconfigured
                ATM cells received.  This entry applies to ATM pports
                only."
        ::={ pportEntry 65 }
```

```
pportAtmNumBitsVCI OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This entry holds a count of the VCI bits supported by this
                ATM card."
        ::={ pportEntry 66 }

pportAtmNumBitsVPI OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This entry holds a count of the VPI bits supppoted my this
                ATM card."
        ::={ pportEntry 67 }

pportAtmInterfaceType OBJECT-TYPE
        SYNTAX  INTEGER {
                uni         (1),
                nni         (2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Integer used to set ATM UNI or NNI type"
        ::= { pportEntry 68 }

pportSonetSDHLoopbackConfig OBJECT-TYPE
        SYNTAX  INTEGER {
                noLoop          (1),
                lineLoop        (2),
                diagnosticsLoop (3)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Integer used to set the loopback state of the Sonet/SDH
                port "
        ::= { pportEntry 69 }

pportSonetSDHLoopStatus OBJECT-TYPE
        SYNTAX INTEGER {
                noLoop  (1),
                lineLoop (2),
                diagnosticsLoop (3),
                metalicLoop     (4),
                noMetalicLoop   (5)
        }
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Integer indicating the current loopback status of the
                Sonet/SDH port"
        ::= { pportEntry 70 }

pportOutDiscardsCell OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of outbound cell discarded due to
                congestion."
        ::= { pportEntry 71 }

pportAtmPlcp OBJECT-TYPE
        SYNTAX  INTEGER {
                enable                (1),
                disable               (2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                " The PLCP maintains the state of the ATM Physical Layer
                Convergance Protocol. When the protocol is disabled atm
                direct mapping is used with hec cell delineation."
        ::={ pportEntry 72 }

pportCbrTargetClockMode OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
               "The clocking method to be used by the CE and structured
                CBR card.

                4-port circuit emulation and CBR cards
                --------------------------------------

                1 - synchronous (external) clocking
                2 - SRTS clocking
                3 - adaptive clock method"
        ::= { pportEntry 73 }
```

```
pportCbrCurrentClockMode OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory

        DESCRIPTION
                "The clocking method currently used by the CBR card.

                4-port CBR card
                -----------------------------------------------
                1 - synchronous (external) clocking
                2 - SRTS clocking
                3 - adaptive clock method"
        ::= { pportEntry 74 }

pportClockMasterChannel OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The number of the channel (lport) on this pport that is
                used as ACM/SRTS master for this line interface:

                    4-port CBR E1/DS1 card (structured mode): 1..4"
        ::= { pportEntry 75 }

pportFibreType OBJECT-TYPE
        SYNTAX  INTEGER {
                sonetMultiMode(4),
                sonetShortSingleMode(2)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The type of fibre connected to the ATM-IWU. Its type
                determines the maximum transmission distance.

                1-port ATM-IWU STM-1/STS-3c card
                --------------------------------
                4 - multimode fibre (max transmission distance approx. 2 km)
                2 - single mode fibre short haul (max transmission
                    distance approx. 20 km)"
        ::= { pportEntry 76 }

pportLaserStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                off (1),
                on (2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
               "The configured state of the laser:

                1-port ATM-IWU STM-1/STS-3c card & Topaz 4-port OC3 & STM-
                1 card
                ------------------------------------------------------------

                1 - off
                2 - on"
        ::= { pportEntry 77 }

pportMaxActiveVpiBits OBJECT-TYPE
        SYNTAX  INTEGER (0..12)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The maximum number of active VPI bits configured for use
                at the ATM interface. At the ATM UNI, the maximum number
                of active VPI bits configured for use ranges from 0 to 8
                only."
        ::= { pportEntry 78 }

pportBipErrorsThresh OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The threshold of BIP errors. If the number of line
                BIP-24/8 errors persists to exceed this value, the pport
                will be taken down and a trap will be issued:

                1-port ATM-IWU STM-1/STS-3c card:
                ---------------------------------
                1 - ignore BIP errors
                4 - 10^-4 errors
                5 - 10^-5 errors
                6 - 10^-6 errors"
        ::= { pportEntry 79 }

pportBipSectionErrors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of B1 section BIP-8 errors since the last
                reset."
        ::= { pportEntry 80 }
pportBipLineErrors OBJECT-TYPE
        SYNTAX  Counter
```

```
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of line BIP-24/8 errors since the last reset."
        ::= { pportEntry 81 }

pportBipPathErrors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUs  mandatory
        DESCRIPTION
                "The number of B3 path BIP-8 errors since the last reset."
        ::= { pportEntry 82 }

pportFebeErrors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of line Far End Block Errors since the last
                reset."
        ::= { pportEntry 83 }

pportHcsErrors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of correctable HCS errors since the last
                reset."
        ::= { pportEntry 84 }

pportHcsSevereErrors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of uncorrectable HCS errors since the last
                reset."
        ::= { pportEntry 85 }

pportCongestedReceivedCells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of received cells that were discarded due to
                congestion of the ATM-IWU, since the last reset."
        ::= { pportEntry 86 }


pportCongestedTransmittedCells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of cells to be transmitted, that were
                discarded due to congestion of the ATM-IWU, since the last
                reset."
        ::= { pportEntry 87 }

pportAtmLayerErroredReceivedCells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of received cells that were discarded due to
                ATM layer errors, since the last reset."
        ::= { pportEntry 88 }
pportAtmLayerErroredTransmittedCells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of cells to be transmitted, that were
                discarded due to ATM layer errors, since the last reset."
        ::= { pportEntry 89 }

pportDS0BitStuff OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Bit stuffing on or off for 2047 bit pattern in DS0 far
                end testing."
        ::= { pportEntry 90 }

pportDS0BitErrorCount OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total number of errors in 2047 DS0 far end testing."
        ::= { pportEntry 91 }

pportDS0BitErrorFreeSeconds OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total number of error free seconds in 2047 DS0 far end
                testing."
        ::= { pportEntry 92 }
```

```
pportDS0BitErroredSeconds OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total number of errored seconds in 2047 DS0 far end
                testing."
        ::= { pportEntry 93 }

pportDS0MidspanRepeaters OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of Midspan repeaters in a link to be punched
                thru for DS0 far end testing."
        ::= { pportEntry 94 }

pportDS0TestPatternSync OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "While DS0 far end testing, the DS0 pattern has detected
                the pattern four consecutive times.  If the pattern is
                lost, this variable will be equal to one."
        ::= { pportEntry 95 }

pportDS0InjectBitError OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "While testing DS0 far end lpbk, setting this variable will
                inject a bit error into the test pattern."
        ::= { pportEntry 96 }

pportDS0FarendLpbkType OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The type of DS0 equipment to be looped back at the far end
                during DS0 far end testing.
                    1 - OCU
                    2 - DSU
                    3 - CSU."
        ::= { pportEntry 97 }

pportDS0LpbkMode OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "This determines what mode the DS0 Processor will be set
                to.
                    1 - no lpbk
                    2 - switch lpbk
                    3 - far end lpbk."
        ::= { pportEntry 98 }

pportDS0SwitchLpbkStart OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "This is the first DS0 in a string of consecutive DS0's to
                be looped back at the switch if the DS0 processor is set
                to switch loopback. If it's set to far end loopback, this
                determines which singleton DS0 will be monitored at the
                receive end."
        ::= { pportEntry 99 }

pportDS0SwitchLpbkEnd OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "This is the Last DS0, +1, in a sting of consecutive DS0's
                to be looped back at the switch if the DS0 processor is
                set to switch loopback. If the DS0 processor is set to far
                end loopback, this value will equal
                pportDS0SwitchLpbkStart + 1.."
        ::= { pportEntry 100 }

pportDS0FarendDS0InLpbk OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS   mandatory
        DESCRIPTION
                "This indicates the DS0 in far end loopback."
        ::= { pportEntry 101 }

pportDS0SendTestTraffic OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Enable or Disable sending DS0 2047 test pattern for DS0
                far end testing."
        ::= { pportEntry 102 }
```

```
pportOc3LoopConfig OBJECT-TYPE
        SYNTAX  INTEGER {
                oc3NoLoop     (1),
                oc3AtmLoop    (2),
                oc3SerPhyLoop (3),
                oc3ParPhyLoop (4)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The loopback state of the ATM IWU OC3"
        ::= { pportEntry 103 }

pportOc3LoopStatus OBJECT-TYPE
        SYNTAX INTEGER {
                oc3NoLoop     (1),
                oc3SerPhyLoop (3), -- internal serial loopback
                oc3ParPhyLoop (4) -- internal parallel loopback
        }
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The current loopback status of the ATM IWU OC3 / ATM-CS"
        ::= { pportEntry 104 }

pportISDNIpBaseAddr OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Specifies the base address used for IP Dynamic Address
                Assignment"
        ::= { pportEntry 105 }

pportSonetSTM1Scramble OBJECT-TYPE
        SYNTAX  INTEGER {
                disabled (1),
                enabled(2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Configured SONET/STM-1 scrambling."
        ::= { pportEntry 106 }

pportEFCIMarking OBJECT-TYPE
        SYNTAX  INTEGER {
                disabled (1),
                enabled(2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Configured EFCI marking for cell traffic in Topaz IOMs"
        ::= { pportEntry 107 }

pportAtmQOSTransmitMode OBJECT-TYPE
        SYNTAX  INTEGER {
                fix-priority (1),
                weighted-round-robin (2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Select fix priority or weighted round-robin for cell
                 transmission in different ATM QOS classes."
        ::= { pportEntry 108 }

pportHECMode OBJECT-TYPE
        SYNTAX INTEGER {
                disabled (1),
                enabled(2)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Configured ATM HEC single bit error correction routine."
        ::= { pportEntry 109 }

pportISDNChannelStatus    OBJECT-TYPE
        SYNTAX OCTET STRING
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The ISDN call status of all the B-Channels on the pport
                are encoded into this 24 character, (for T1 ISDN PRI
                card...23B+D) or 31 character,
                (for E1 ISDN PRI card...30B+D ) ASCII string. The
                respective call state for
                each B-Channel is represented in its corresponding bit
                position. The D-Channel status is similarly encoded into
                the appropriate bit position (24th for T1) or
                (16th for E1). The encoding legend is as follows...

                I B-Channel is in Idle state
                R B-Channel is in Dialing (Ringing) state
                C B-Channel is in Connected (Active) state
                H B-Channel is in Releasing (Hanging-up) state
                U D-Channel is Up
```

```
                D D-Channel is Down"
        ::= { pportEntry 110 }

pportds1FarEndLoopStatus OBJECT-TYPE
        SYNTAX INTEGER {
                fe_noloop        (1),
                fe_line_loop     (2),
                 fe_payload_loop  (3)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "DS1 Far End Loopback Status as commanded by the switch"
        ::= { pportEntry 111 }

pportds1FDLControl OBJECT-TYPE
        SYNTAX INTEGER {
                disabled (1),
                enabled  (2)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Enables or Disables FDL Function for DS1 ESF pport"
        ::= { pportEntry 112 }

pportds1FDLPrmXmit OBJECT-TYPE
        SYNTAX INTEGER {
                disabled (1),
                enabled  (2)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Enables or Disables transmission of Performance
                Monitoring Report (PRM) messages on the DS1 ESF FDL"
        ::= { pportEntry 113 }

pportds1FDLPidXmit OBJECT-TYPE
        SYNTAX INTEGER {
                disabled (1),
                enabled  (2)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Enables or Disables transmission of Path ID (PID)
                messages on the DS1 ESF FDL"
        ::= { pportEntry 114 }

pportds1FDLXmitPid OBJECT-TYPE
        SYNTAX OCTET STRING
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Transmit Path Id to be sent on the DS1 ESF FDL.  (76
                bytes)"
        ::= { pportEntry 115 }

pportds1FDLRcvPid OBJECT-TYPE
        SYNTAX OCTET STRING
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The last path id message received on the DS1 ESF FDL.
                (76 bytes)"
        ::= { pportEntry 116 }

pportds1FDLRcvTsid OBJECT-TYPE
        SYNTAX OCTET STRING
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The last test id message received on the DS1 ESF FDL.
                (76 bytes)"
        ::= { pportEntry 117 }
```

```
pportSonetSDHFramingMode OBJECT-TYPE
        SYNTAX  INTEGER {
                sonet (1),
                sdh (2)
        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The Framing mode for SONET/SDH port interface types."
        ::= { pportEntry 118 }

pportds1InbandLoopType OBJECT-TYPE
        SYNTAX INTEGER {
                ds1CSU (1),
                ds1NI (2)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Integer used to distinguish Inband Line loopback code sent
                from the switch when commanding inband line loopbacks,and
                reckognized by the switch when responding to remote
                loopback commands"
        ::= { pportEntry 119 }

pportESFDataLinkStatus OBJECT-TYPE
         SYNTAX  INTEGER {
                inService    (1),
                outOfService (2)
        }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The current status of the DS1 ESF (FDL) data link."
        ::= { pportEntry 120 }

pportPMTcaId OBJECT-TYPE
        SYNTAX  INTEGER {
                currentThresholdCVL   (1),
                currentThresholdESL   (2),
                currentThresholdSESL  (3),
                currentThresholdUASL  (4),
                currentThresholdCVP   (5),
                currentThresholdESP   (6),
                currentThresholdSESP  (7),
                currentThresholdSASP  (8),
                currentThresholdCSSP  (9),
                currentThresholdUASP  (10),
                currentThresholdCVS   (11),
                currentThresholdESS   (12),
                currentThresholdSESS  (13),
                currentThresholdESx   (14),
                dayThresholdCVL       (15),
                dayThresholdESL       (16),
                dayThresholdSESL      (17),
                dayThresholdUASL      (18),
                dayThresholdCVP       (19),
                dayThresholdESP       (20),
                dayThresholdSESP      (21),
                dayThresholdSASP      (22),
                dayThresholdCSSP      (23),
                dayThresholdUASP      (24),
                dayThresholdCVS       (25),
                dayThresholdESS       (26),
                dayThresholdSESS      (27),
                dayThresholdESx       (28),
                currentThresholdCVCPP (29),
                currentThresholdESCPP (30),
                currentThresholdSESCPP(31),
                currentThresholdSASCPP(32),
                currentThresholdUASCPP(33),
                dayThresholdCVCPP     (34),
                dayThresholdESCPP     (35),
                dayThresholdSESCPP    (36),
                dayThresholdSASCPP    (37),
                dayThresholdUASCPP    (38),
                currentThresholdCVCPPFE (39),
                currentThresholdESCPPFE (40),
                currentThresholdSESCPPFE(41),
                currentThresholdSASCPPFE(42),
                currentThresholdUASCPPFE(43),
                dayThresholdCVCPPFE     (44),
                dayThresholdESCPPFE     (45),
                dayThresholdSESCPPFE    (46),
                dayThresholdSASCPPFE    (47),
                dayThresholdUASCPPFE    (48)
        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "This object identifies the the most recently declared
                pport threshold crossing alert"
        ::= { pportEntry 121 }

pportBchanTimerValue OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                 "This value is used on a per port basis to add a delay to
                  answering a ISDN call of the initial Disconnect from the
                  network.  This due to the ARP cache not updating quick
                  enough on the routers"
         ::= { pportEntry 122 }
```

```
pportAAL5CRC32Error OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Number of received AAL5 packets with CRC32 errors."
        ::= { pportEntry 123 }

pportAAL5CPIError OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Number of received AAL5 packets with CPI errors
                The only valid value currently defined for the CPI
                field is all zeros"
        ::= { pportEntry 124 }

pportAAL5LengthError OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
        "Number of received AAL5 packets which satisfied one of the
        following error conditions:
                1. number of received cells * 48 bytes - length value in
                trailer > 55 bytes
                2. number of received cells * 48 bytes - length value in
                trailer < 8 bytes"
        ::= { pportEntry 125 }

pportAAL5ReassemblyTimerError OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Number of expired reassembly timers"
        ::= { pportEntry 126 }

pportAAL5MaxNrSegError OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Number of received AAL5 packets which exceeds the maximum
                allowed length"
        ::= { pportEntry 127 }

pportRedundancyArch OBJECT-TYPE
        SYNTAX INTEGER {
                disabled (1),
                one-plus-one-aps (2)
        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                 "Pport redundancy architecture.  Default is disabled."
         ::= { pportEntry 128 }

pportAPSadminDir OBJECT-TYPE
        SYNTAX INTEGER {
                uni-directional (1),
                bi-directional (2)
        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                 "Provisioned APS switch-direction-mode. Default is uni-
                directional."
         ::= { pportEntry 129 }

pportAPSlineType OBJECT-TYPE
        SYNTAX INTEGER {
                working-line (1),
                protection-line (2)
        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                 "APS line type for the pport.  Writable only when
                pportRedundancyArch is disabled."
         ::= { pportEntry 130 }

pportAPSrevertiveMode OBJECT-TYPE
        SYNTAX INTEGER {
                revertive (1),
                nonrevertive (2)
        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                 "APS revertive mode.  When revertive, after the condition
                for an automatic switchover clears, user traffic will be
                switched back to the working line after the
                pportAPSwtrPeriod expires. Default is revertive."
         ::= { pportEntry 131 }
```

```
pportAPSpairedSlotId OBJECT-TYPE
        SYNTAX INTEGER (1..16)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Slot ID of the paired-with APS pport.  Writable only
                when pportRedundancyArch is disabled."
         ::= { pportEntry 132 }

pportAPSpairedPportId OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Pport ID of the paired-with APS pport.  Writable only
                when pportRedundancyARch is disabled."
         ::= { pportEntry 133 }

pportAPSsfBerThresh OBJECT-TYPE
        SYNTAX INTEGER (3..5)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Exponent N for APS Signal Failure based on BER.  A
                line bit-error-rate above 10^-N causes an SF BER failure
                to be asserted.  SF BER is cleared when the line BER
                returns to less than 10^-7."
         ::= { pportEntry 134 }

pportAPSsdBerThresh OBJECT-TYPE
        SYNTAX INTEGER (6..9)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                 "Exponent N for APS Signal Degrade based on BER.  A line
                bit-error-rate above 10^-N causes an SD BER failure to be
                asserted.  SD BER is cleared when the line BER returns to
                less than 10^-(N+1)."
         ::= { pportEntry 135 }

pportAPSwtrPeriod OBJECT-TYPE
        SYNTAX INTEGER (5..12)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                 "APS wait-to-restore period.  The number of minutes to
                wait after an automatic switch condition clears before
                switching back to the working line."
         ::= { pportEntry 136 }

pportAPSprotectionLineState OBJECT-TYPE
        SYNTAX INTEGER {
                released (1),
                selected (2)
        }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                 "Indicates the state of the protection line selector.
                When selected, a protection switchover has taken place
                and user traffic is being selected from the protection
                line."
         ::= { pportEntry 137 }

pportAPSxCommand OBJECT-TYPE
        SYNTAX INTEGER {
                clear (1),
                lockout-protection (2),
                forced-switch-w-to-p (3),
                forced-switch-p-to-w (4),
                manual-switch-w-to-p (5),
                manual-switch-p-to-w (6),
                exercise (7)
        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                 "External switch commands.  The clear command clears
                any previously activated external command.  Manual
                switch is preemptable by auto-switch-requests (line
                failures) while forced switch is not preemptable (except
                if the protection line fails).  Manual and forced switch
                from protection to working, is valid only for 1+1 mode.
                Exercise simulates a switchover using APS signalling
                without actually performing a switch to protection line."
         ::= { pportEntry 138 }
```

```
pportAPSconfigStatus OBJECT-TYPE
        SYNTAX INTEGER {
                not-configured (1),
                valid (2),
                invalid (3)
        }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                 "Status of APS pport configuration.  The not-configured
                status indicates the pport APS feature is not configured
                or it is a non-APS pport.  The valid state indicates the
                pport APS function is configured and has been validated
                by the APS manager.  The invalid state indicates that the
                APS manager has detected a conflict between the working
                line and protection line pport configurations.  The user
                must clear the invalid state by changing the configuration
                of the misconfigured pport as soon as possible.  Check
                pportAPSadminDir, pportAPSlineType, pportAPSrevertiveMode,
                and pportAPSwtrPeriod for any misconfigurations."
         ::= { pportEntry 139 }

pportAPSOperRxStatus OBJECT-TYPE
        SYNTAX INTEGER {
                up (1),
                down (2),
                testing (3)
        }
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The current state of the pair of APS pports with respect
                to the ability of the pair to Receive user traffic.  When
                indicated as up, user traffic may be received on at least
                one of the ports in the APS pair."
        ::= { pportEntry 140 }
```

Traffic Shaper Parameter Table

```
pportTrafficShaperTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF PportTrafficShaperEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A list of Parameters assigned to the 16 traffic shaper
                parameter conbinations of each pport of the ATM-IWU."
        ::= { pport 3 }

pportTrafficShaperEntry OBJECT-TYPE
        SYNTAX  PportTrafficShaperEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "Information of a single system reference clock source."
        INDEX   { pportSlotId, pportId, pportTrafficShaperIndex }
        ::= { pportTrafficShaperTable 1 }

PportTrafficShaperEntry ::=
        SEQUENCE {
                pportTrafficShaperIndex
                    INTEGER,
                pportTrafficShaperPriority
                    INTEGER,
                pportTrafficShaperCellRatioDividend
                    INTEGER,
                pportTrafficShaperCellRatioDivisor
                    INTEGER,
                pportTrafficShaperPeak
                    INTEGER,
                pportTrafficShaperCredit
                    INTEGER
        }

pportTrafficShaperIndex OBJECT-TYPE
        SYNTAX  INTEGER (1..16)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "A positive integer to identify an entry in the
                pportTrafficShaperTable.  The ATM-IWU and CS cards support
                exactly 16 shapers."
        ::= { pportTrafficShaperEntry 1 }

pportTrafficShaperPriority OBJECT-TYPE
        SYNTAX  INTEGER (0..15)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The priority of the Shaper.  Channels tied to a Shaper
                are served only if no higher priority Shapers await
                service. 0 is the highest, 15 the lowest priority."
        ::= { pportTrafficShaperEntry 2 }
```

```
pportTrafficShaperCellRatioDividend OBJECT-TYPE
        SYNTAX  INTEGER (0..255)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The quotient of pportTrafficShaperCellRatioDividend and
                pportTrafficShaperCellRatioDivisor determines the average
                rate allocated to the shaper. The shaper serves
                portTrafficShaperCellRatioDividend cells in
                pportTrafficShaperCellRatioDivisor cell times."
        ::= { pportTrafficShaperEntry 3 }

pportTrafficShaperCellRatioDivisor OBJECT-TYPE
        SYNTAX  INTEGER (1.. 16777215)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "See pportTrafficShaperCellRatioDividend"
        ::= { pportTrafficShaperEntry 4 }

pportTrafficShaperPeak OBJECT-TYPE
        SYNTAX  INTEGER (0..255)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The peak cell rate.
                It defines the minimum gap (in cell units) between
                emission of any consecutive cells in this channel."
        ::= { pportTrafficShaperEntry 5 }

pportTrafficShaperCredit OBJECT-TYPE
        SYNTAX  INTEGER (0..255)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Defines the maximum number of credits that can be
                accumulated by a shaper. This is equivalent to the maximum
                burst allowed at the peak rate."
        ::= { pportTrafficShaperEntry 6 }
```

The Physical Channel Group

```
--      The variables that configure physical channels on a port at a node

channelNumber OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of physical channels (regardless of their
                current state present at this node."
        ::= { chan 1 }

channelTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF ChanEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A list of physical channel entries.  The number of
                entries is given by the value of channelNumber."
        ::= { chan 2 }

chanEntry OBJECT-TYPE
        SYNTAX  ChanEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "The physical channel entry contains objects relevant to a
                physical channel."
        INDEX   { chanSlotId, chanPortId,
                  chanId }
        ::= { channelTable 1 }

ChanEntry ::=
        SEQUENCE {
                chanSlotId
                    INTEGER,
                chanPortId
                    INTEGER,
                chanId
                    INTEGER,
                chanAdminType
                    INTEGER,
                chanNumLport
                    INTEGER,
                chanDataRate
                    INTEGER,
                chanType
                    INTEGER,
                chanXmitClock
                    INTEGER,
                chanAdminStatus
                    INTEGER,
                chanOperStatus
                    INTEGER,
                chanDs1LineType
                    INTEGER,
                chanDs1ZeroCoding
                    INTEGER,
                chanInOctets
                    Counter,
                chanInFrames
                    Counter,
                chanInDiscards
                    Counter,
                chanInErrors
                    Counter,
```

```
                chanOutOctets
                    Counter,
                chanOutFrames
                    Counter,
                chanOutDiscards
                    Counter,
                chanOutErrors
                    Counter,
                chanBertPattern
                    INTEGER,
                chanBertUserBytes
                    INTEGER,
                chanBertErrorRate
                    INTEGER,
                chanBertCommand
                    INTEGER,
                chanBertStatus
                    INTEGER,
                chanBertBitCount
                    Gauge,
                chanBertErrorCount
                    Gauge,
                chanLinkDownReason
                    INTEGER,
                chands1SendFarEndCode
                    INTEGER,
                chands1CodeTypeRsp
                    INTEGER,
                chands1NearEndLoopConfig
                    INTEGER,
                chands1NearEndLoopStatus
                    INTEGER,
                chanBackupClock
                    INTEGER,
                chanFT1DS0s
                    INTEGER
                }

chanSlotId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The slot number of the corresponding physical channel."
        ::= { chanEntry 1 }

chanPortId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The port number of this channel on the board."
        ::= { chanEntry 2 }

chanId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The channel number of the corresponding physical channel."
        ::= { chanEntry 3 }

chanAdminType OBJECT-TYPE
        SYNTAX  INTEGER {
                ft3-1 (9)       -- 1-port Fractional T3 i/o card
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The defined type of the board which the physical channel
                is on."
        ::= { chanEntry 4 }

chanNumLport OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of logical ports on the physical channel."
        ::= { chanEntry 5 }

chanDataRate OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "An estimate of the physical channels's data rate in bits
                per second. The data rate can't be to an arbitrary value.
                The legal values are for CHANDS3 are 1344000, 1536000"
        ::= { chanEntry 6 }

chanType OBJECT-TYPE
        SYNTAX  INTEGER {
                ft3-1 (9)       -- 1-port Fractional T3 i/o card
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The actual type of the physical channel."
        ::= { chanEntry 7 }
```

```
chanXmitClock OBJECT-TYPE
        SYNTAX  INTEGER {
                loopTimed(1),
                internal(2),
                external(3)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The transmit clock source."
        ::= { chanEntry 8 }

chanAdminStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                    up (1),
                    down (2),
                    testing (3),
                    invalid (255)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The desired state of the physical channel."
        ::= { chanEntry 9 }

chanOperStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                    up (1),
                    down (2),
                    testing (3)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The current state of the physical channel."
        ::= { chanEntry 10 }

chanDs1LineType OBJECT-TYPE
        SYNTAX  INTEGER {
                    d4 (1),
                    esf (2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The line type of the T1 or FT1-24B channel. ANSI ESF is
                equivalent to Bellcore ESF."
        ::= { chanEntry 11 }

chanDs1ZeroCoding OBJECT-TYPE
        SYNTAX  INTEGER {
                    nx64 (1),
                    nx56 (2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The variety of zero code suppression used on the other end
                 of the T1 link will change the information rate available"
        ::= { chanEntry 12 }

chanInOctets OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of octets received on the physical
                channel, including framing characters."
        ::= { chanEntry 13 }

chanInFrames OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of frames received on the physical
                channel."
        ::= { chanEntry 14 }

chanInDiscards OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of inbound frames discarded."
        ::= { chanEntry 15 }

chanInErrors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of inbound frames that contained
                erroneous headers (e.g., illegal or unknown DLCIs)."
        ::= { chanEntry 16 }
```

```
chanOutOctets OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of octets transmitted out of the physical
                channel, including framing characters."
        ::= { chanEntry 17 }

chanOutFrames OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of frames transmitted out of the physical
                channel."
        ::= { chanEntry 18 }

chanOutDiscards OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of outbound frames discarded due to
                severe congestion."
        ::= { chanEntry 19 }

chanOutErrors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of outbound frames that causes xmit
                errors."
        ::= { chanEntry 20 }

chanBertPattern OBJECT-TYPE
        SYNTAX  INTEGER {
                allZeros        (1),
                allOnes         (2),
                oneZero         (3),
                oneOneZeroZero  (4),
                oneOf8          (5),
                threeOf24       (6),
                qRSS            (7),
                user1Byte       (8),
                user2Byte       (9),
                user3Byte       (10),
                user4Byte(      11)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Pattern generated by the XBERT"
        ::= { chanEntry 21 }

chanBertUserBytes OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Definition of 1,2,3 or 4 byte pattern if UserNByte
                selected."
        ::= { chanEntry 22 }

chanBertErrorRate OBJECT-TYPE
        SYNTAX  INTEGER {
                none            (1),
                tenMinusThree(2),
                tenMinusSix(3)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Error rate to insert in generated pattern."
        ::= { chanEntry 23 }

chanBertCommand OBJECT-TYPE
        SYNTAX  INTEGER {
                start           (1),
                stop            (2),
                clearCounters(3),
                injectError(4)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Single shot commands to the BERT."
        ::= { chanEntry 24 }

chanBertStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                unused          (1),
                unavailable     (2),
                outOfFrame      (3),
                inFrame         (4)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current status of the BERT for this channel."
        ::= { chanEntry 25 }
```

```
chanBertBitCount OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of bits received - stops counting at
                0xFFFFFFFF"
        ::= { chanEntry 26 }

chanBertErrorCount OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of bits received in error- stops counting
                 at 0xFFFFFFFF"
        ::= { chanEntry 27 }

chanLinkDownReason OBJECT-TYPE
        SYNTAX  INTEGER {
                none            (0),
                red-alarm       (1),
                yellow-alarm    (2),
                blue-alarm      (4),
                carrier-loss    (8),
                looped-back     (16)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Reason why the link is down.  The blue-alarm is equivalent
                to the Alarm Indication Signal (AIS) failure."
        ::= { chanEntry 28 }
```

```
chands1SendFarEndCode OBJECT-TYPE
        SYNTAX INTEGER {
                ds1SendNoCode           (1),
                ds1SendFramedCSULineActuateLoop      (2),
                ds1SendFramedCSULineReleaseLoop      (3),
                ds1SendFramedNILineActuateLoop       (4),
                ds1SendFramedNILineReleaseLoop       (5),
                ds1SendFdlESFAnsiLineActuateLoop     (6),
                ds1SendFdlESFAnsiLineReleaseLoop     (7),
                ds1SendFdlESFAnsiPayloadActuateLoop  (8),
                ds1SendFdlESFAnsiPayloadReleaseLoop  (9),
                ds1SendUnframedCSULineActuateLoop    (10),
                ds1SendUnframedCSULineReleaseLoop    (11),
                ds1SendUnframedNILineActuateLoop     (12),
                ds1SendUnframedNILineReleaseLoop     (13),
                ds1SendOOBNILineActuateLoop          (14),
                ds1SendOOBNILineReleaseLoop          (15)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Integer used to send loopback code to far-end DS1"
        ::= { chanEntry 29 }

chands1CodeTypeRsp OBJECT-TYPE
        SYNTAX INTEGER {
                ds1CSUFramedInbandLineLoop   (1),
                ds1NNIFramedInbandLineLoop   (2)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Integer used to distinguish Framed Inband Line loopback
                code
                (CSU or NNI) that near end will recognise from far-end DS1"
        ::= { chanEntry 30 }

chands1NearEndLoopConfig OBJECT-TYPE
        SYNTAX INTEGER {
                ds1ClearLoop   (1),
                ds1PayloadLoop (2),
                ds1LineLoop    (3),
                ds1DiagLoop    (4)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Integer used to set the near end loopback state of the
                DS1, Diag is xmit => recv"
        ::= { chanEntry 31 }
```

```
chands1NearEndLoopStatus OBJECT-TYPE
        SYNTAX INTEGER {
                noloop                  (1),
                framedInbandLineLoop    (2),
                fdlESFAnsiLineLoop      (3),
                fdlESFAnsiPayloadLoop   (4),
                dS3CbitLineLoop         (5),
                nMSLineLoop             (6),
                nMSPayloadLoop          (7),
                nMSDiagLoop             (8)
                }
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Integer indicating the current loopback status of the near
                 end of the DS1"
        ::= { chanEntry 32 }

chanBackupClock OBJECT-TYPE
        SYNTAX  INTEGER {
                loop            (1),
                internal(2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Backup clock source if External timing fails."
        ::= { chanEntry 33 }

chanFT1DS0s OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "DS0 usage bitmask for Fractional T1 data rates."
        ::= { chanEntry 34 }
```

The Logical Port Group

```
--  The variables that configure logical ports at a node

lportTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF LportEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A list of logical port entries.  The number of entries is
                given by the value of ifNumber in MIB-II."
        ::= { lport 1 }

lportEntry OBJECT-TYPE
        SYNTAX  LportEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "The logical port entry contains objects relevant to a
                logical port."
        INDEX   { lportIfIndex }
        ::= { lportTable 1 }

LportEntry ::=
        SEQUENCE {
                lportIfIndex
                    Index,
                lportSlotId
                    INTEGER,
                lportPportId
                    INTEGER,
                lportId
                    INTEGER,
                lportLink
                    INTEGER,
                lportProtocol
                    INTEGER,
                lportSignal
                    INTEGER,
                lportFt1Ds0s
                    DisplayString,
                lportGlobalDlci
                    INTEGER,
                lportDlcmiStd
                    INTEGER,
                lportDlciAddrFmt
                    INTEGER,
                lportDlciAddrLen
                    INTEGER,
                lportMaxFramesize
                    INTEGER,
                lportDceVerifTimer
                    Counter,
                lportDceErrorThresh
                    Counter,
                lportDceEventCount
                    Counter,
                lportDteErrorThresh
                    Counter,
                lportDteEventCount
                    Counter,
                lportDtePollTimer
                    Counter,
                lportDteFullCounter
                    Counter,
                lportDteMulticast
                    INTEGER,
```

```
lportTrkRnode
    IpAddress,
lportTrkRlport
    INTEGER,
lportCongestState
    INTEGER,
lportQP1Len
    INTEGER,
lportQP2Len
    INTEGER,
lportQP3Len
    INTEGER,
lportQP4Len
    INTEGER,
lportErrTime
    TimeTicks,
lportErrType
    INTEGER,
lportErrData
    OCTET STRING,
lportDiagTestId
    INTEGER,
lportDiagTestRuns
    INTEGER,
lportDataRate
    INTEGER,
lportTrkStatus
    INTEGER,
lportSevCongests
    INTEGER,
lportAbsCongests
    INTEGER,
lportTrkOverhead
    INTEGER,
lportTrkUtil
    INTEGER,
lportVAvailbw
    INTEGER,
lportTrkLastTimeChange
    TimeTicks,
lportCongestRate
    INTEGER,
lportCongestRateThresh
    INTEGER,
lportDiagState
    INTEGER,
lportDiagOptionStr
    OCTET STRING,
lportDiagPassCount
    INTEGER,
lportDiagFailCount
    INTEGER,
lportDiagResultStr
    DisplayString,
lportDs0BitStuff
    INTEGER,
lportErrorThreshold
    INTEGER,
lportUnsyncBandwidth
    INTEGER,
lportDTEInStatusFrames
    Counter,
lportDTEInFullStatusFrames
    Counter,
lportDTEInAsyncStatusFrames
    Counter,
lportDTEInErrorFrames
    Counter,
lportDTEOutPollFrames
    Counter,
lportDTEPollErrorCounts
    Counter,
lportDTEFailCounts
    Counter,
lportDTEFailReason
    INTEGER,
lportDTEOperStatus
    INTEGER,
lportDCEInPollFrames
    Counter,
lportDCEInErrorFrames
    Counter,
lportDCEOutStatusFrames
    Counter,
lportDCEOutFullStatusFrames
    Counter,
lportDCEOutAsyncStatusFrames
    Counter,
lportDCEPollErrorCounts
    Counter,
lportDCEFailCounts
    Counter,
lportDCEFailReason
    INTEGER,
lportDCEOperStatus
    INTEGER,
lportDCEOperDlcmiStd
    INTEGER,
lportLMIInErrorFrames
    Counter,
lportDCEnN4
    INTEGER,
lportDCEnT3
    INTEGER,
lportXmitLatencyThreshold
    INTEGER,
lportXmitRefillPriority0Percentage
    INTEGER,
```

```
lportXmitRefillPriority1Percentage
    INTEGER,
lportXmitRefillPriority2Percentage
    INTEGER,
lportXmitRefillPriority3Percentage
    INTEGER,
lportAbsoluteThreshold
    INTEGER,
lportSevereThreshold
    INTEGER,
lportMildThreshold
    INTEGER,
lportAtmUPCEnable
    INTEGER,
lportAtmUniType
    INTEGER,
lportConnectionType
    INTEGER,
lportAtmCellType
    INTEGER,
lportTrkKeepAliveTimer
    INTEGER,
lportTrkKeepAliveErrorThreshold
    INTEGER,
lportIgCutThruStatus
    INTEGER,
lportEgCutThruStatus
    INTEGER,
lportEgCutThruThresh
    INTEGER,
lportFrameRelayTrkEnable
    INTEGER,
lportSmdsSsiIf
    INTEGER,
lportSmdsSsiSlot
    INTEGER,
lportSmdsScrnId
    INTEGER,
lportSmdsIaScrnOp
    INTEGER,
lportSmdsGaScrnOp
    INTEGER,
lportSmdsIaScrnMap
    OCTET STRING,
lportSmdsGaScrnMap
    OCTET STRING,
lportSmdsTrkAddr
    OCTET STRING,
lportSmdsCrc
    INTEGER,
lportSmdsCpePoll
    INTEGER,
lportSmdsPduCheck
    INTEGER,
lportSmdsCntOutFrDxi2HbPolls
    Counter,
lportSmdsCntOutByteDxi2HbPolls
    Counter,
lportSmdsCntInFrDxi2HbPolls
    Counter,
lportSmdsCntInByteDxi2HbPolls
    Counter,
lportSmdsCntOutFrSip3s
    Counter,
lportSmdsCntOutByteSip3s
    Counter,
lportSmdsCntInFrSip3s
    Counter,
lportSmdsCntInByteSip3s
    Counter,
lportSmdsCntDxi2LinkIdInvalids
    Counter,
lportSmdsCntDxi2StationIdInvalids
    Counter,
lportSmdsCntDxi2CrInvalids
    Counter,
lportSmdsCntDxi2AeInvalids
    Counter,
lportSmdsCntDxi2CtrlInvalids
    Counter,
lportSmdsCntDxi2FrameSizeErrors
    Counter,
lportSmdsCntSip3RsvdInvalids
    Counter,
lportSmdsCntSip3BetagMismatchs
    Counter,
lportSmdsCntSip3BasizeIncorrects
    Counter,
lportSmdsCntSip3BasizeInvalids
    Counter,
lportSmdsCntSip3DaTypeInvalids
    Counter,
lportSmdsCntSip3DaInvalids
    Counter,
lportSmdsCntSip3SaTypeInvalids
    Counter,
lportSmdsCntSip3SaInvalids
    Counter,
lportSmdsCntSip3BasizeMismatchs
    Counter,
lportSmdsCntSip3HeLenInvalids
    Counter,
lportSmdsCntSip3HeVersionInvalids
    Counter,
lportSmdsCntSip3HeCarrierInvalids
    Counter,
lportSmdsCntSip3Crc32Errors
    Counter,
```

```
lportSmdsCntSip3TRsvdInvalids
    Counter,
lportSmdsCntSaNotFounds
    Counter,
lportSmdsCntSaValidationFails
    Counter,
lportSmdsCntSaDaOnSamePorts
    Counter,
lportSmdsCntDaSsiMismacths
    Counter,
lportSmdsCntSsiProvisionErrors
    Counter,
lportSmdsCntDstIaNotFounds
    Counter,
lportSmdsCntDstGaNotFounds
    Counter,
lportSmdsCntSrcIaScrnFails
    Counter,
lportSmdsCntDstIaScrnFails
    Counter,
lportSmdsCntDstGaScrnFails
    Counter,
lportSmdsTotalDiscards
    Counter,
lportSmdsSsiNode
    INTEGER,
lportBilling
    INTEGER,
lportSmdsCntDstGaSrcIsCascade
    INTEGER,
lportLinkStatus
    INTEGER,
lportLMIDelay
    INTEGER,
lportCRC
    INTEGER,
lportSmdsMulticastGa
    OCTET STRING,
lportSmdsMulticastIpAddr
    IpAddress,
lportAtmVPI
    INTEGER,
lportAtmVCI
    INTEGER,
lportPeakCellRateindex
    INTEGER,
lportSustCellRate
    INTEGER,
lportBurstTolerance
    INTEGER,
lportBuTrkOnFailure
    INTEGER,
lportTrkFailureThrsh
    INTEGER,
lportTrkRestThrsh
    INTEGER,
lportBuTrkRetryInt
    INTEGER,
lportBuTrkRetryNum
    INTEGER,
lportBuTrkCycleInt
    INTEGER,
lportTrkManualBu
    INTEGER,
lportPrimTrk
    Index,
lportInitCallSetup
    INTEGER,
lportBuFailReason
    INTEGER,
lportQ922Enable
    INTEGER,
lportQ922State
    INTEGER,
lportTrkPduRevision
    INTEGER,
lportPVCMgrPduRevision
    INTEGER,
lportDS0LoopStatus
    INTEGER,
lportISDNDuration
    INTEGER,
lportISDNSourceAddr
    OCTET STRING,
lportISDNDestAddr
    OCTET STRING,
lportISDNIpAddr
    IpAddress,
lportISDNCallRejCause
    INTEGER,
lportLastInvalidDLCI
    INTEGER,
lportTrkProtState
    INTEGER,
lportTrkTrafficMix
    INTEGER,
lportNumVC
    INTEGER,
lportTrkAdminCost
    INTEGER,
lportPrivateNet
    INTEGER,
lportTrkStaticDelay
    INTEGER,
lportTrkDynamicDelay
    INTEGER,
lportAtmDataRateQoS1
    INTEGER,
```

```
lportAtmDataRateQoS2
    INTEGER,
lportAtmDataRateQoS3
    INTEGER,
lportAtmDataRateQoS4
    INTEGER,
lportOutVAvailbwQoS1
    INTEGER,
lportOutVAvailbwQoS2
    INTEGER,
lportOutVAvailbwQoS3
    INTEGER,
lportOutVAvailbwQoS4
    INTEGER,
lportInVAvailbwQoS1
    INTEGER,
lportInVAvailbwQoS2
    INTEGER,
lportInVAvailbwQoS3
    INTEGER,
lportInVAvailbwQoS4
    INTEGER,
lportOutAllocbwQoS1
    INTEGER,
lportOutAllocbwQoS2
    INTEGER,
lportOutAllocbwQoS3
    INTEGER,
lportOutAllocbwQoS4
    INTEGER,
lportInAllocbwQoS1
    INTEGER,
lportInAllocbwQoS2
    INTEGER,
lportInAllocbwQoS3
    INTEGER,
lportInAllocbwQoS4
    INTEGER,
lportDynamicQoSbw
    INTEGER,
lportSvcHoldDownTimer
    INTEGER,
lportAtmConnectionType
    INTEGER,
lportAtmRouteMetricQoS1
    INTEGER,
lportAtmRouteMetricQoS2
    INTEGER,
lportAtmRouteMetricQoS3
    INTEGER,
lportAtmRouteMetricQoS4
    INTEGER,
lportIlmiPollTimeout
    INTEGER,
lportAtmProtocol
    INTEGER,
lportIlmiAdminStatus
    INTEGER,
lportIlmiOperStatus
    INTEGER,
lportIlmiPollRetry
    INTEGER,
lportAtmVpiBits
    INTEGER,
lportAtmVciBits
    INTEGER,
lportAtmOamAlarmEnable
    INTEGER,
    -- 218 is deprecated
    -- this OID will be reused
lportbwUNIPolicy
    INTEGER,
lportStarvation
    INTEGER,
lportRecOverflow
    INTEGER,
lportLossOfCellSequence
    INTEGER,
lportLossOfStructurePointer
    INTEGER,
lportCbrInsDummyCells
    Counter,
lportRecFifoUnderflowCnt
    Counter,
lportRecFifoOverflowCnt
    Counter,
lportCbrLossOfStructurePointerCnt
    Counter,
lportCbrLossOfCellSequenceCnt
    Counter,
lportShaperId
    INTEGER,
lportDteIlmiPrefixScreenMode
    INTEGER,
lportSmdsPduViolTca
    INTEGER,
lportSmdsPduViolThresh
    INTEGER,
lportSmdsPduHdrSip3SaNotFound
    OCTET STRING,
lportSmdsPduHdrSip3SaDaOnSamePort
    OCTET STRING,
lportSmdsPduHdrSip3DstGaNotFound
    OCTET STRING,
lportSmdsPduHdrSip3DstIaScrnFail
    OCTET STRING,
lportSmdsPduHdrSip3SaValFail
    OCTET STRING,
```

```
lportSmdsPduHdrSip3DstIaNotFound
    OCTET STRING,
lportSmdsPduHdrSip3SrcIaScrnFail
    OCTET STRING,
lportSmdsPduHdrSip3DstGaScrnFail
    OCTET STRING,
lportSmdsPduHdrSip3SaTypeInvalid
    OCTET STRING,
lportSmdsPduHdrSip3DaTypeInvalid
    OCTET STRING,
lportSmdsPduHdrDxi2LinkIdInvalid
    OCTET STRING,
lportSmdsPduHdrDxi2CrInvalid
    OCTET STRING,
lportSmdsPduHdrDxi2CtrlInvalid
    OCTET STRING,
lportSmdsPduHdrDxi2StationIdInvalid
    OCTET STRING,
lportSmdsPduHdrDxi2AeInvalid
    OCTET STRING,
lportDS0FarendLpbkEnabled
    INTEGER,
lportDS0FarendLpbkDisabled
    INTEGER,
lportTrkProtFailureThreshold
    INTEGER,
lportPtr
    OCTET STRING,
lportISDNPoolUtil
    INTEGER,
lportPPPNegotiationFailCode
    INTEGER,
lportTrkUtilQoS1
    INTEGER,
lportTrkUtilQoS2
    INTEGER,
lportTrkUtilQoS3
    INTEGER,
lportTrkUtilQoS4
    INTEGER,
lportIlmiNumOctetsTx
    INTEGER,
lportIlmiNumOctetsRx
    INTEGER,
lportIlmiNumPdusTx
    INTEGER,
lportIlmiNumPdusRx
    INTEGER,
lportIlmiNumErrorsRx
    INTEGER,
lportIlmiNumUmePollsTx
    INTEGER,
lportIlmiNumUmeResponsesRx
    INTEGER,
lportIlmiVPI
    INTEGER,
lportIlmiVCI
    INTEGER,
lportInCells
    Counter,
lportOutCells
    Counter,
lportDS1ChannelId
    INTEGER,
lportCDV
    INTEGER,
lportAtmTrkIomCktDiagStr
    OCTET STRING,
lportAtmTrkSpCktDiagStr
    OCTET STRING,
lportAuthState
    INTEGER,
lportAuthDomainID
    INTEGER,
lportAuthPPPOption
    INTEGER,
lportAuthFailReason
    INTEGER,
lportEchoRequestOption
    INTEGER,
lportEchoRequestInterval
    INTEGER,
lportEchoRequestMaxTries
    INTEGER,
lportMultilinkProtocolOption
    INTEGER,
lportMultilinkProtocolFailReason
    INTEGER,
lportBandwidthAllocProtocolOption
    INTEGER,
lportBandwidthAllocProtocolCallFailReason
    INTEGER,
lportPrivateNetOverflow
    INTEGER,
lportCbrFifoHalfLength
    INTEGER,
lportCustomerID
    INTEGER,
lportCongestThresh0
    INTEGER,
lportCongestThresh1
    INTEGER,
lportCongestThresh2
    INTEGER,
lportCongestThresh3
    INTEGER,
lportSevereCongestNotifyTime
    INTEGER,
```

```
lportSevereCongestStatus
    INTEGER,
 lportSmdsNumInFramesIa
     Counter,
 lportSmdsNumInBytesIa
      Counter,
 lportSmdsNumInFramesGa
      Counter,
 lportSmdsNumInBytesGa
      Counter,
 lportSmdsNumOutFramesIa
       Counter,
 lportSmdsNumOutBytesIa
       Counter,
 lportSmdsNumOutFramesGa
       Counter,
lportSmdsNumOutBytesGa
      Counter,
 -- 301, 302, 303, 304 are available
lportBadPVCFactor
    INTEGER,
lportAmberReductionPm
    INTEGER,
lportAmberReductionPs
    INTEGER,
lportCongestionCheckInterval
    INTEGER,
lportCongestionClearDelay
    INTEGER,
lportNrtsRmGenType
    INTEGER,
lportNrtsRmTermType
    INTEGER,
lportNrtsEfciCheck
    INTEGER,
lportNrtsBufAlloc
    INTEGER,
lportNrtsClp01Thresh
    INTEGER,
lportNrtsDiscardThresh
    INTEGER,
lportNrtsEfciThresh
    INTEGER,
lportNrtsRmCellCount
    Counter,
lportCloseLoopSwitch
    INTEGER,
lportAtmVPIStop
    INTEGER,
lportServiceClassType
    INTEGER,
lportSegmentation
    INTEGER,
lportTrkOSPFAreaID
        IpAddress,
    lportMultiServiceCBR
        INTEGER,
    lportMultiServiceVBRrt
        INTEGER,
    lportMultiServiceVBRnrt
        INTEGER,
    lportMultiServiceUBR
        INTEGER
}

lportIfIndex OBJECT-TYPE
    SYNTAX  Index
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The ifIndex value of the corresponding ifEntry."
    ::= { lportEntry 1 }

lportSlotId OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The slot number of the board the port is on."
    ::= { lportEntry 2 }

lportPportId OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The physical port number of the interface on the board."
    ::= { lportEntry 3 }
```

```
lportId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The logical port number of the interface on the physical
                port. A logical port can be uniquely identified in a
                Cascade network as NodeId.lportSlotId.lportPportId.lportId.
                lportId is defaulted to 1 (i.e., one-to-one mapping
                between the physical port and the logical port such as
                local V.35 or carrier) except the following cases:

                - For ufr, nfr and trk logical port, it indicates the
                  bundle number for fractional T1 or 24-bundle T1.
                - For pdntrk logical port, it indicates the dlci over the
                  PDN.

                Note that pdntrk is not allowed on fractional T1 or
                24-bundle T1 physical port."
        ::= { lportEntry 4 }

lportLink OBJECT-TYPE
        SYNTAX  INTEGER {
                user (0),       -- user link: connecting to non-casc
                trk (1),        -- trunk: connecting to casc switch
                transport (2)   -- transport: connecting 2 FR networks
                                   using one circuit
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The equipment type of the peer to which the logical port
                is connected."
        ::= { lportEntry 5 }

lportProtocol OBJECT-TYPE
        SYNTAX  INTEGER {
                    fr (0),                 -- frame relay protocol
                    nfr (1),                -- non-frame relay protocol
                                            -- (direct frad usr port or casc
                                               trunk)
                    fradPPPto1294 (2),      -- xlation frad, PPP-to-1294
                    smds(3),                -- SMDS
                    atm (4),                -- ATM user-network
                    isdndchan (5),          -- ISDN pri d-channel
                    dirmgmttrk(7),          -- Direct management trunk for
                                               SMDS
                    smdsoptmgmt(8)          -- SMDS OPT management trunk
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The data link protocol running on the logical port."
        ::= { lportEntry 6 }

lportSignal OBJECT-TYPE
        SYNTAX  INTEGER {
                    dce (1),    -- network side
                    dte (2),    -- user side
                    nni (3)     -- bi-directional, both network & user
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The user-network signalling capability for SVC and DLCMI
                of the logical port.
                Note that this only applies to fr protocol port."
        ::= { lportEntry 7 }

lportFt1Ds0s OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The bit mask indicating the DS0s for the fractional T1/E1
                logical port which must be a subset of the corresponding
                pportDs1Ds0s. It's represented by a 32-bit hex char
                string."
        ::= { lportEntry 8 }

lportGlobalDlci OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The global DLCI correspondent to the interface if the
                DLCI is globally significant in the network. Note that
                this object is read-write only during creation, and read-
                only after creation."
        ::= { lportEntry 9 }

portDlcmiStd OBJECT-TYPE
        SYNTAX  INTEGER {
                not-applicable (0), -- Not applicable, for example, on a
                                       trunk
                disabled       (1),
                lmiRev1        (2),
                ansiT1-617-D   (3), -- ANSI T1.617 Annex D
                ccittQ-933-A   (4), -- CCITT Q.933 Annex A
                autodetect     (5), -- Auto Detection
                ansiT1-617-B   (6)  -- ANSI T1.617 Annex B
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "This variable states which Data  Link  Connection
                Management scheme is active (and by implication, what DLCI
                it uses) on the Frame Relay interface."
        ::= { lportEntry 10 }
```

```
lportDlciAddrFmt OBJECT-TYPE
        SYNTAX  INTEGER {
                q922            (1)  -- 10-bit (CCITT Standard)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This variable states which address format is in use on the
                Frame Relay interface."
        ::= { lportEntry 11 }

lportDlciAddrLen OBJECT-TYPE
        SYNTAX  INTEGER {
                two-octets-10-bits (1),    -- Current Standard
                three-octets-10-bits (2),
                three-octets-16-bits (3),
                four-octets-17-bits (4),
                four-octets-23-bits (5)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This variable states which address length in octets. In
                the case of Q922 format, the length indicates the entire
                length of the address including the control portion."
        ::= { lportEntry 12 }

lportMaxFramesize OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The maximum frame size supported on the interface.
                Currently it's not used."
        ::= { lportEntry 13 }

lportDceVerifTimer OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The polling verification timer (secs) on the DCE
                interface. If the port is configured as a DXI/SNI, then
                this MIB object defines the heartbeat poll timer setting.
                This value must be between 5 and 30."
        ::= { lportEntry 14 }

lportDceErrorThresh OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The error threshold on the DCE interface.
                 If the port is configured as a DXI/SNI, then this MIB
                object defines the heartbeat poll No Ack threshold setting.
                 This value must be between 1 and 10
                 when the interface is not a DXI/SNI."
        ::= { lportEntry 15 }

lportDceEventCount OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The monitored events count on the DCE interface.
                 This value must be between 1 and 10."
        ::= { lportEntry 16 }

lportDteErrorThresh OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The error threshold on the DTE interface.
                 This value must be between 1 and 10."
        ::= { lportEntry 17 }

lportDteEventCount OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The monitored events count on the DTE interface.
                 This value must be between 1 and 10."
        ::= { lportEntry 18 }

lportDtePollTimer OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The link status polling timer (secs) on the DTE interface.
                 This value must be between 5 and 30."
        ::= { lportEntry 19 }
```

```
lportDteFullCounter OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The full status polling counter on the DTE interface.
                 This value must be between 1 and 255."
        ::= { lportEntry 20 }

lportDteMulticast OBJECT-TYPE
        SYNTAX  INTEGER {
                    one-way (1),
                    two-way (2),
                    m-way (3)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The multicast capability of the interface."
        ::= { lportEntry 21 }

lportTrkRnode OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Node at the other end of the trunk. This object is read-
                write only during lport creation, and read-only
                thereafter."
        ::= { lportEntry 22 }

lportTrkRlport OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Logical port (IfIndex) at the other end of the trunk."
        ::= { lportEntry 23 }

lportCongestState OBJECT-TYPE
        SYNTAX  INTEGER {
                    normal (1),   -- below "knee" point
                    mild (2),     -- between "knee" & "cliff" points
                    severe (3),   -- above "cliff" point
                    critical (4) -- xmit queue is full
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Congestion State for the logical port."
        ::= { lportEntry 24 }

lportQP1Len OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Priority-1 queue length (number of packets) in xmit
                buffer."
        ::= { lportEntry 25 }

lportQP2Len OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Priority-2 queue length (number of packets) in xmit
                buffer."
        ::= { lportEntry 26 }

lportQP3Len OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Priority-3 queue length (number of packets) in xmit
                buffer."
        ::= { lportEntry 27 }

lportQP4Len OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Priority-4 queue length (number of packets) in xmit
                buffer."
        ::= { lportEntry 28 }

lportErrTime OBJECT-TYPE
        SYNTAX  TimeTicks
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The value of sysUpTime at which the last error was
                detected."
        ::= { lportEntry 29 }
```

```
lportErrType OBJECT-TYPE
        SYNTAX  INTEGER {
                  short-frame             (1),
                  hdlc-abort              (2),
                  residual-bit            (3),
                  crc-Error               (4),
                  rcv-Long                (5),
                  rcv-overrun             (6),
                  tx-underrun             (7),
                  unknownError            (8),
                  illegalDLCI             (9),
                  unknownDLCI             (10),
                  dlcmiProtoErr           (11),
                  dlcmiUnknownIE          (12),
                  dlcmiSequenceErr        (13),
                  dlcmiUnknownRpt         (14),
                  unknownProt             (15),
                  discardFW               (16),
                  discardRange            (17),
                  discardPortMismatch     (18),
                  discardIllegalLen       (19)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The type of error that was last seen on this interface."
        ::= { lportEntry 30 }

lportErrData OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "An octet string containing as much of the error packet as
                possible.  As a minimum, it must contain the Q.922 Address
                or as much as was delivered.  It is desirable to include
                all information up to the PDU."
        ::= { lportEntry 31 }

lportDiagTestId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Identification for the diagnostics tests to be run."
        ::= { lportEntry 32 }

lportDiagTestRuns OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of passes of the diagnostics tests to be run.
                The default value is 1."
        ::= { lportEntry 33 }

lportDataRate OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "An estimate of the logical port's data rate in bits per
                second."
        ::= { lportEntry 34 }

lportTrkStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                    ndown (0),
                    nattempt (1),
                    ninit (2),
                    n2way (3),
                    nexstart (4),
                    nexchange (5),
                    nloading (6),
                    nfull (7),
                    btdefined (9)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The current state of the trunk.  The btdefined state only
                 applies to backup trunks."
        ::= { lportEntry 35 }

lportSevCongests OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The cumulative number of times that the lport's congestion
                state has changed from mildly-congested to severely-
                congested since the last reset."
        ::= { lportEntry 36 }
```

```
lportAbsCongests OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The cumulative number of times that the lport's congestion
                state has changed from severely-congested to absolutely-
                congested since the last reset."
        ::= { lportEntry 37

lportTrkOverhead OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "An estimate of the overhead (headers and internal control
                messages) in trunking user data in terms of a percentage of
                the configured trunk bandwidth."
        ::= { lportEntry 38 }

lportTrkUtil OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "An estimate of the real utilization of the trunk bandwidth
                in terms of a percentage of the configured trunk
                bandwidth."
        ::= { lportEntry 39 }

lportVAvailbw OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Currently available virtual trunk bandwidth."
        ::= { lportEntry 40 }


lportTrkLastTimeChange OBJECT-TYPE
        SYNTAX  TimeTicks
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The sysUpTime value when the trunk was last changed to
                the current status."
        ::= { lportEntry 41 }

lportCongestRate OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The rate (%) of entering severely or absolutely congested
                states in the last one minute interval."
        ::= { lportEntry 42 )

lportCongestRateThresh OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The alert threshold (%) for the CongestRate; trap will be
                sent when exceeded."
        ::= { lportEntry 43 }

lportDiagState OBJECT-TYPE
        SYNTAX  INTEGER {
                inactive (0),
                active(1)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The current state of the diagnostic on this logical port."
        ::= { lportEntry 44 }

lportDiagOptionStr OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Optional parameters to the diagnostic."
        ::= { lportEntry 45 }

lportDiagPassCount OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Number of successful diagnostic passes."
        ::= { lportEntry 46 }

lportDiagFailCount OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Number of failed diagnostic passes."
        ::= { lportEntry 47 }
```

```
lportDiagResultStr OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Description of last diagnostic failure."
        ::= { lportEntry 48 }

lportDs0BitStuff OBJECT-TYPE
        SYNTAX  INTEGER {
                no-bit-stuffing (0),
                bit-stuffing (1)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Is bit stuffing enabled on this lport?"
        ::= { lportEntry 49 }

lportErrorThreshold OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The per minute error threshold before a link error
                trap is sent.  Rounded down to nearest power of 2.
                Value of 0 = link error traps never sent."
        ::= { lportEntry 50 }

lportUnsyncBandwidth OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The accumulated allocated/deallocated bandwidth which
                has not been propagated by OSPF yet."
        ::= { lportEntry 51 }

lportDTEInStatusFrames OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of LMI STATUS frames received from the DCE
                since the last system reset. The count includes link
                integrity verification frames only."
        ::= { lportEntry 52 }

lportDTEInFullStatusFrames OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of LMI Full STATUS frames received from the
                DTE since the last system reset."
        ::= { lportEntry 53 }

lportDTEInAsyncStatusFrames OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of asynchronous LMI Status frames received
                from the DTE since the last system reset. For LMI Rev 1,
                these are Update STATUS frames. For ANSI Annex D and CCITT
                Annex A these are Asynchronous STATUS frames."
        ::= { lportEntry 54 }

lportDTEInErrorFrames OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of LMI frames received containing protocol
                errors."
        ::= { lportEntry 55 }

lportDTEOutPollFrames OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of LMI Status Enquiry frames transmitted since
                the last system reset."
        ::= { lportEntry 56 }

lportDTEPollErrorCounts OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of DTE in-channel signaling link reliability
                errors (i.e. LMI Status Enquiry frames that were not
                responded to, sequence number errors) since the last system
                reset."
        ::= { lportEntry 57 }
```

```
lportDTEFailCounts OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of times since the last system reset LMI has
                declared the DTE side of the link down due to excessive
                errors."
        ::= { lportEntry 58 }

lportDTEFailReason OBJECT-TYPE
        SYNTAX  INTEGER {
                   ok          (0), -- no failure
                   dte-bad-Nr  (1), -- received Nr != Ns
                   dte-timeout (2), -- timeout waiting for STATUS message
                   prot-error  (3)  -- protocol error
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The LMI failure reason. If the rate of LMI detected
                errors exceeds the threshold, this contains the cause of
                the most recent error."
        ::= { lportEntry 59 }

lportDTEOperStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                   invalid (0),
                   up      (1),
                   down    (2)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The operational status of the DTE side of LMI on this
                link."
        ::= { lportEntry 60 }

lportDCEInPollFrames OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of LMI poll frames received from the DTE since
                the last system reset."
        ::= { lportEntry 61 }

lportDCEInErrorFrames OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of LMI frames received containing protocol
                errors."
        ::= { lportEntry 62 }

lportDCEOutStatusFrames OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of LMI STATUS frames transmitted since the
                last system reset. The count includes link integrity
                verification frames only."
        ::= { lportEntry 63 }

lportDCEOutFullStatusFrames OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of LMI Full STATUS frames transmitted since the
                last system reset."
        ::= { lportEntry 64 }

lportDCEOutAsyncStatusFrames OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of asynchronous LMI Status frames transmitted
                since the last system reset. For LMI Rev 1, these are
                Update STATUS frames. For ANSI Annex D and CCITT Annex A
                these are Asynchronous STATUS frames."
        ::= { lportEntry 65 }

lportDCEPollErrorCounts OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of DCE in-channel signaling link reliability
                errors (i.e. timeouts waiting for LMI Status Enquiry
                frames, sequence number errors) since the last system
                reset."
        ::= { lportEntry 66 }
```

```
lportDCEFailCounts OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of times since the last system reset LMI has
                declared the DCE side of the link down due to excessive
                errors."
        ::= { lportEntry 67 }

lportDCEFailReason OBJECT-TYPE
        SYNTAX INTEGER {
                ok              (0), -- no failure
                dce-bad-Nr      (1), -- received Nr != Ns
                dce-timeout     (2), -- timeout waiting for Status Enquiry
                                     -- message
                prot-error      (3)  -- protocol error
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The LMI failure reason. If the rate of LMI detected
                errors exceeds the threshold, this contains the cause of
                the most recent error."
        ::= { lportEntry 68 }

lportDCEOperStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                   invalid (0),
                   up      (1),
                   down    (2)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The operational status of the DCE side of LMI on this
                link."
        ::= { lportEntry 69 }

lportDCEOperDlcmiStd OBJECT-TYPE
        SYNTAX  INTEGER {
                  unknown      (1),
                  lmiRev1      (2),
                  ansiT1-617-D (3),  -- ANSI T1.617 Annex D
                  ccittQ-933-A (4),  -- CCITT Q.933 Annex A
                  reserved     (5),
                  ansiT1-617-B (6)   -- ANSI T1.617 Annex B
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This variable states which Data Link Connection Management
                scheme is actually being run on this link. This is used
                for DCE links configured for AutoDetect and indicates the
                LMI standard used in the last poll received from the DTE."
        ::= { lportEntry 70 }

lportLMIInErrorFrames OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of total frames received with an error. For
                NNI links, this is the sum of lportDTEInErrorFrames and
                lportDCEInErrorFrames."
        ::= { lportEntry 71 }

lportDCEnN4 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "This variable specifies the maximum number of LMI Status
                Enquiry frames that can be received from a DTE within time
                interval lportDCEnT3. This is only valid on lports using
                LMIREV1."
        ::= { lportEntry 72 }

lportDCEnT3 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "This variable specifies the time interval (secs) used to
                determine if lportDCEnN4 an excess number (lportDCEnN4) of
                Status Enquiry messages are received. This is only valid
                on lports using LMIREV1."
        ::= { lportEntry 73 }
```

```
lportXmitLatencyThreshold OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Outbound frames are allowed on the transmit commit queue
                when the commit queue falls below this threshold (in
                microseconds)."
        ::= { lportEntry 74 }

lportXmitRefillPriority0Percentage OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Percentage of time the transmit queue is refilled in
                priority order 0, 1, 2, 3."
        ::= { lportEntry 75 }

lportXmitRefillPriority1Percentage OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Percentage of time the transmit queue is refilled in
                priority order 1, 2, 3, 0."
        ::= { lportEntry 76 }

lportXmitRefillPriority2Percentage OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Percentage of time the transmit queue is refilled in
                priority order 2, 3, 0, 1."
        ::= { lportEntry 77 }

lportXmitRefillPriority3Percentage OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Percentage of time the transmit queue is refilled in
                priority order 3, 0, 1, 2."
        ::= { lportEntry 78 }

lportAbsoluteThreshold OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Absolute Congestion Threshold.  When the time average
                queue length (TAQL) or the absolute queue length hits
                1. (Severe-congestion-threshold + 3/4 (Absolute-congestion-
                    threshold - Severe-congestion-threshold)), light
                    absolute congestion state is indicated or
                2. (Absolute-congestion-threshold), heavy absolute
                    congestion state is indicated.
                This threshold value is configured in units of 56 byte
                buffers."
        ::= { lportEntry 79 }

lportSevereThreshold  OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
        "Severe congestion threshold.  When the time average queue length
        (TAQL) hits
        1. 1/2(Mild-congestion-threshold + Severe-congestion-threshold),
        light severe congestion state is indicated or
        2. (Severe-congestion-threshold), heavy severe congestion state
        is indicated.
        This threshold value is configured in units of 56 byte buffers."
        ::= { lportEntry 80 }

lportMildThreshold  OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Mild congestion threshold. When the time average queue
                length (TAQL) hits
                1. 3/4 (Mild-congestion-threshold), light-mild congestion
                state is indicated or
                2. (Mild-congestion-threshold), heavy-mild congestion
                state is indicated.
                This threshold value is configured in units of 56 byte
                buffers."
        ::= { lportEntry 81 }
```

```
lportAtmUPCEnable  OBJECT-TYPE
       SYNTAX  INTEGER {
               disabled(1),
               enabled (2)
       }
       ACCESS  read-write
       STATUS  mandatory
       DESCRIPTION
       "Enable ATM UPC Function."
       ::= { lportEntry 82 }

lportAtmUniType  OBJECT-TYPE
       SYNTAX INTEGER {
               public  (1),
               private  (2)
       }
 ACCESS read-write
 STATUS mandatory
 DESCRIPTION
       "The type of UNI for this ATM lport."
       ::= { lportEntry 83 }

lportConnectionType  OBJECT-TYPE
       SYNTAX INTEGER {
               network-endsystem  (1),
               network-network  (2)
       }
 ACCESS read-write
 STATUS mandatory
 DESCRIPTION
       "The type of connection at this lport."
       ::= { lportEntry 84 }

lportAtmCellType  OBJECT-TYPE
       SYNTAX INTEGER {
               atm-uni-cell-hdr  (1),
               atm-nni-cell-hdr (2)
       }
 ACCESS read-write
 STATUS mandatory
 DESCRIPTION
       "atm cell header with uni-format or nni-format with no GFC."
       ::= { lportEntry 85 }

lportTrkKeepAliveTimer OBJECT-TYPE
       SYNTAX  INTEGER
       ACCESS  read-write
       STATUS  mandatory
       DESCRIPTION
               "Send a keep-alive packet after this many seconds.  These
               packets are sent on trunk links only.  The range is 0 thru
               15, with zero meaning the keep-alive protocol is
               disabled.  The default is one."
       ::= { lportEntry 86 }

lportTrkKeepAliveErrorThreshold OBJECT-TYPE
       SYNTAX  INTEGER
       ACCESS  read-write
       STATUS  mandatory
       DESCRIPTION
               "Declare the interface down after this many keep-alive
               packets in a row are missed.  The default is five."
       ::= { lportEntry 87 }

lportIgCutThruStatus OBJECT-TYPE
       SYNTAX  INTEGER {
                    disabled  (1),
                    enabled   (2)
               }
       ACCESS  read-write
       STATUS  mandatory
       DESCRIPTION
               "Enable ingress cut-thru on this user lport.  Default is
               disabled."
       ::= { lportEntry 88 }

lportEgCutThruStatus OBJECT-TYPE
       SYNTAX  INTEGER {
                    disabled  (1),
                    enabled   (2)
               }
       ACCESS  read-write
       STATUS  mandatory
       DESCRIPTION
               "Enable egress cut-thru on this user lport.  Default is
               disabled."
       ::= { lportEntry 89 }

lportEgCutThruThresh OBJECT-TYPE
       SYNTAX  INTEGER
       ACCESS  read-write
       STATUS  mandatory
       DESCRIPTION
               "Egress cut thru threshold - start transmitting a frame on
               a user port after this many segments have arrived."
       ::= { lportEntry 90 }

lportFrameRelayTrkEnable OBJECT-TYPE
       SYNTAX  INTEGER {
                    allowed   (1),
                    not-allowed(2)
               }
       ACCESS  read-write
       STATUS  mandatory
       DESCRIPTION
               "Allow Frame Relay Trunks on the user link."
       ::= { lportEntry 91 }
```

```
lportSmdsSsiIf OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The interface number of the SSI to which this DXI/SNI is
                multiplexed."
        ::= { lportEntry 92 }

lportSmdsSsiSlot OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The slot of the SSI to which this DXI/SNI is multiplexed."
        ::= { lportEntry 93 }

lportSmdsScrnId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The ID of address screen for this interface."
        ::= { lportEntry 94 }

lportSmdsIaScrnOp OBJECT-TYPE
        SYNTAX  INTEGER {
                     allow     (1),
                     disallow  (2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Individual addrtess screen operation."
        ::= { lportEntry 95 }

lportSmdsGaScrnOp OBJECT-TYPE
        SYNTAX  INTEGER {
                     allow     (1),
                     disallow  (2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Group address screen operation."
        ::= { lportEntry 96 }

lportSmdsIaScrnMap OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The individual address screen member bit map. For set
                request, the first byte is the operation : 1 for deleting,
                2 for adding. The following bytes are the bit map. The
                bit-position of each bit in this bit map represent a screen
                member ID. The most significant bit is corresponding to
                group address ID 1. For get response, The whole
                string is the bit map."
        ::= { lportEntry 97 }

lportSmdsGaScrnMap OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The group address screen member bit map. For set request,
                the first byte is the operation : 1 for deleting, 2 for
                adding. The following bytes are the bit map. The bit
                position of each bit in this bit map represent a screen
                member ID. The most significant bit is corresponding to
                group address ID 1. For get response, The whole
                string is the bit map."
        ::= { lportEntry 98 }

lportSmdsTrkAddr OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "SMDS address. For SSI feeder port, this address is the
                local address of all SMDS optimim paths associated with
                this SSI feeder port. For SSI Optimum path port, this
                address is the remote address of this SSI optimum path.
                The 4 most significant bits are 1100. The following 4 bits
                are 0001. The following 5 bytes are the 10 digits number
                in BCD format. The following 16 bits are padded with 1's"
        ::= { lportEntry 99 }

lportSmdsCrc OBJECT-TYPE
        SYNTAX  INTEGER {
                     crc16     (1),
                     crc32     (2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "CRC which CPE generates."
        ::= { lportEntry 100 }
```

```
lportSmdsCpePoll OBJECT-TYPE
        SYNTAX  INTEGER {
                    nopoll    (1),
                    poll      (2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Whether CPE supports heart beat poll or not."
        ::= { lportEntry 101 }

lportSmdsPduCheck OBJECT-TYPE
        SYNTAX  INTEGER {
                    off (0),
                    on  (1)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Turn on/off the detailed SIP3 PDU error checking."
        ::= { lportEntry 102 }

lportSmdsCntOutFrDxi2HbPolls OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of DXI2 heart beat poll frames transmitted."
        ::= { lportEntry 103 }

lportSmdsCntOutByteDxi2HbPolls OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Total number of octets in DXI2 heart beat poll frames
                transmitted."
        ::= { lportEntry 104 }

lportSmdsCntInFrDxi2HbPolls OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of DXI2 heart beat poll frames received."
        ::= { lportEntry 105 }

lportSmdsCntInByteDxi2HbPolls OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Total number of octets in DXI2 heart beat poll frames
                received."
        ::= { lportEntry 106 }

lportSmdsCntOutFrSip3s OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of SIP3 frames transmitted."
        ::= { lportEntry 107 }

lportSmdsCntOutByteSip3s OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Total number of octets in SIP3 frames transmitted."
        ::= { lportEntry 108 }

lportSmdsCntInFrSip3s OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of SIP3 frames received."
        ::= { lportEntry 109 }

lportSmdsCntInByteSip3s OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Total number of octets in SIP3 frames received."
        ::= { lportEntry 110 }

lportSmdsCntDxi2LinkIdInvalids OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of instances that DXI2 link ID is invalid."
        ::= { lportEntry 111 }
```

```
lportSmdsCntDxi2StationIdInvalids OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of instances that DXI2 station ID is invalid."
        ::= { lportEntry 112 }

lportSmdsCntDxi2CrInvalids OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of instances that DXI2 command/response field is
                invalid."
        ::= { lportEntry 113 }

lportSmdsCntDxi2AeInvalids OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of instances that DXI2 address extension field is
                invalid."
        ::= { lportEntry 114 }

lportSmdsCntDxi2CtrlInvalids OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of instances that DXI2 control field is invalid."
        ::= { lportEntry 115 }

lportSmdsCntDxi2FrameSizeErrors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of instances there is a DXI2 frame size error."
        ::= { lportEntry 116 }
lportSmdsCntSip3RsvdInvalids OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of instances that SIP3 reserved field in header is
                invalid."
        ::= { lportEntry 117 }

lportSmdsCntSip3BetagMismatchs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of instances that SIP3 BE tag is not macthed."
        ::= { lportEntry 118 }

lportSmdsCntSip3BasizeIncorrects OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of instances that SIP3 BA size is incorrect."
        ::= { lportEntry 119 }

lportSmdsCntSip3BasizeInvalids OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of instances that SIP3 BA size is invalid."
        ::= { lportEntry 120 }

lportSmdsCntSip3DaTypeInvalids OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of instances that SIP3 destination address type is
                invalid."
        ::= { lportEntry 121 }

lportSmdsCntSip3DaInvalids OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of instances that SIP3 destination address is
                invalid."
        ::= { lportEntry 122 }

lportSmdsCntSip3SaTypeInvalids OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of instances that SIP3 source address type is
                invalid."
        ::= { lportEntry 123 }
```

```
lportSmdsCntSip3SaInvalids OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of instances that SIP3 source address is invalid."
        ::= { lportEntry 124 }

lportSmdsCntSip3BasizeMismatchs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of instances that SIP3 BA sizes in the header and
                trailer are not matched."
        ::= { lportEntry 125 }

lportSmdsCntSip3HeLenInvalids OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of instances that SIP3 header extension length is
                invalid."
        ::= { lportEntry 126 }

lportSmdsCntSip3HeVersionInvalids OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of instances that SIP3 header extension version is
                invalid."
        ::= { lportEntry 127 }

lportSmdsCntSip3HeCarrierInvalids OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of instances that SIP3 header extension carrier is
                invalid."
        ::= { lportEntry 128 }

lportSmdsCntSip3Crc32Errors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of SIP3 CRC errors."
        ::= { lportEntry 129 }

lportSmdsCntSip3TRsvdInvalids OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of instances that SIP3 reserved field in trailer
                is invalid."
        ::= { lportEntry 130 }

lportSmdsCntSaNotFounds OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of instances that source address not found."
        ::= { lportEntry 131 }

lportSmdsCntSaValidationFails OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of instances that source address validation
                failed."
        ::= { lportEntry 132 }

lportSmdsCntSaDaOnSamePorts OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of instances that source address and destination
                addresses are on the same port."
        ::= { lportEntry 133 }

lportSmdsCntDaSsiMismacths OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Obsolete."
        ::= { lportEntry 134 }

lportSmdsCntSsiProvisionErrors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Obsolete."
        ::= { lportEntry 135 }
```

```
lportSmdsCntDstIaNotFounds OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of instances that destination individual address
                not found."
        ::= { lportEntry 136 }

lportSmdsCntDstGaNotFounds OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of instances that destination group address not
                found."
        ::= { lportEntry 137 }

lportSmdsCntSrcIaScrnFails OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of instances that source individual address
                screening failed."
        ::= { lportEntry 138 }

lportSmdsCntDstIaScrnFails OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of instances that destination individual address
                screening failed."
        ::= { lportEntry 139 }

lportSmdsCntDstGaScrnFails OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of instances that destination group address
                screening failed."
        ::= { lportEntry 140 }

lportSmdsTotalDiscards OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Total number of discards."
        ::= { lportEntry 141 }

lportSmdsSsiNode OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The Node of the SSI to which this DXI/SNI is multiplexed."
        ::= { lportEntry 142 }

lportBilling OBJECT-TYPE
        SYNTAX INTEGER {
                    disabled (1),
                    enabled  (2)
                    }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object defines the ability to enable and disable
                billing on this logical port (for SMDS this must be a
                DXI).  When the value of nodeBilling is 'enabled', the
                value of this object
                will take precedence.  When the value of nodeBilling is
                'disabled', the value of this object will be overridden and
                billing will be disabled.

                The default value of this object is 'disabled'.
                "
        ::= {lportEntry 143 }

lportSmdsCntDstGaSrcIsCascade OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This counter is applicable when the netSmdsTrafficMode
                (i.e.; SMDS Group Address processing mode) is Cascade and
                the SMDS lport is SSI. This counter indicates the number
                of instances that a Ga frame was received with a Cascade
                src address.
                "
        ::= { lportEntry 144 }

lportLinkStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                up (1),
                down (2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Link protocol status. The Link protocol may be frame
                relay LMI, DXI heart beat poll, PPP LCP, etc, depends on
                the lport type."
        ::= { lportEntry 145 }
```

```
lportLMIDelay OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The number of seconds 1-9 to buffer (or initiate) a FR
                LMI async update (or ATM OAM alarms) to allow filtering of
                LMI async events. Zero (0) indicates that no buffering is
                done (all updates are iimmediate) and 255 indicates that
                no updates are sent."
        ::= { lportEntry 146 }

lportCRC OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Lport CRC selection:
                 0 - 16 bit CRC
                 1 - 32 bit CRC"
        ::= { lportEntry 147 }

lportSmdsMulticastGa OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The Multicast Group Address is the smds broadcast address
                which is used to get smds address from its IP address."
        ::= { lportEntry 148 }

lportSmdsMulticastIpAddr OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "This IP address is this logical port's IP address."
        ::= { lportEntry 149 }

lportAtmVPI OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "VPI value in the ATM cell header:
                   ATM DXI with HSSI IOP VPI (4 lsb bit) range: 0 - 15
                   ATM UNI DS3/E3 IOP    VPI (4 lsb bit) range: 0 - 15"
        ::= { lportEntry 150 }

lportAtmVCI OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "VCI value in the ATM cell header:
                  ATM DXI with HSSI IOP VCI (6 lsb bit) range: 0 - 63
                  ATM UNI DS3/E3 IOP    VCI (8 lsb bit) range: 0 - 255"
        ::= { lportEntry 151 }

lportPeakCellRateindex OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Peak cell rate queue index"
        ::= { lportEntry 152 }

lportSustCellRate OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Sustainable cell rate in cell/second"
        ::= { lportEntry 153 }

lportBurstTolerance OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Burst Tolerance in multiple of 32 cells"
        ::= { lportEntry 154 }

lportBuTrkOnFailure OBJECT-TYPE
        SYNTAX  INTEGER {
                disabled   (0),
                enabled    (1)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Enables or disables trunk backup due to link down."
        ::= { lportEntry 155 }
```

```
lportTrkFailureThrsh OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The number of seconds a primary trunk must remain down for
                trunk backup to be initiated."
        ::= { lportEntry 156 }

lportTrkRestThrsh OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The number of seconds a primary trunk must remain up for
                trunk backup to be terminated"
        ::= { lportEntry 157 }

lportBuTrkRetryInt OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The number of seconds between backup trunk call setup
                retries."
        ::= { lportEntry 158 }

lportBuTrkRetryNum OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The number of backup trunk call setup tries per call
                setup cycle."
        ::= { lportEntry 159 }

lportBuTrkCycleInt OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The number of seconds between backup trunk call setup
                cycles."
        ::= { lportEntry 160 }

lportTrkManualBu OBJECT-TYPE
        SYNTAX  INTEGER {
                none            (0),
                initCmd         (1),
                termCmd         (2),
                initSched       (3),
                termSched       (4)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Initiate or terminate manual backup for a primary trunk.
                   none         neither initiate or terminate
                   initCmd      initiate operator commanded backup
                   termCmd      terminate operator commanded backup
                   initSched    initiate scheduled backup
                   termSched    terminate scheduled backup"
        ::= { lportEntry 161 }

lportPrimTrk OBJECT-TYPE
        SYNTAX  Index
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Indicates the primary trunk lportIfIndex for which a
                backup trunk is serving as a backup.  This value is also
                used to determine trunk type by the following convention:
                   Value                            Trunk Type
                     0                              Normal
                     lportIfIndex of this trunk     Primary
                     other lportIfIndex             Backup"
        ::= { lportEntry 162 }

lportInitCallSetup OBJECT-TYPE
        SYNTAX  INTEGER {
                  false (0),
                  true (1)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "TRUE if node must initiate backup trunk call setup for
                this primary trunk.  Otherwise FALSE."
        ::= { lportEntry 163 }
```

```
lportBuFailReason OBJECT-TYPE
        SYNTAX  INTEGER {
                  none          (0),
                  buTrkNotDef   (1),
                  buTrkNotEstab(2)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Indicates the reason for the failure to perform trunk
                backup."
        ::= { lportEntry 164 }

lportQ922Enable OBJECT-TYPE
         SYNTAXINTEGER {
                enable (1),
                disable (2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Set 1 to eanble Q.922 on this port. Q.922 must be enabled
                if the port is used for SVC."
         ::= { lportEntry 165 }

lportQ922State OBJECT-TYPE
        SYNTAX  INTEGER {
                uninitialized (0),
                tei-unassigned (1),
                assign-awaiting (2),
                establish-awaiting (3),
                tei-assigned (4),
                awaiting-establishment (5),
                awaiting-release (6),
                multiple-frame-established (7),
                timer-recovery (8)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Q.922 state. This object only makes sense when Q.922 is
                enabled."
        ::= { lportEntry 166 }

lportTrkPduRevision OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The Trunk PDU Revision number being used over this trunk."
        ::= { lportEntry 167 }

lportPVCMgrPduRevision OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The PVC Manager PDU Revision number being used over this
                trunk."
        ::= { lportEntry 168 }

lportDS0LoopStatus OBJECT-TYPE
        SYNTAX  INTEGER {
            normal(1),
            switchlpbk(2),
            farendlpbk(3),
            t1nods0lpbk (4)
        }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Indicates the DS0 Lpbk Status of the Lport."
        ::= { lportEntry 169 }

lportISDNDuration OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of seconds that the ISDN call has been
                established."
        ::= { lportEntry 170 }

lportISDNSourceAddr OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The E.164 address of the source of this ISDN connection."
        ::= { lportEntry 171 }

lportISDNDestAddr OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The E.164 address of the destination of this ISDN
                connection."
        ::= { lportEntry 172 }
```

```
lportISDNIpAddr OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                    "The Ip Address of the client connected to this B-
                    Channel"
        ::= { lportEntry 173 }

lportISDNCallRejCause OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                    "The cause of the call rejection - inability to
                    authenticate or pool is busy."
        ::= { lportEntry 174 }

lportLastInvalidDLCI OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This item pertains to user frame relay lports only.  The
                frame relay header of the received frame contains a dlci,
                which uniquely identifies a specific pvc on this lport.
                When no pvc has been configured, on the lport, that
                corresponds to the dlci specified in the frame header, the
                frame is said to have an invalid dlci.  This lport entry
                holds the value of the most recent invalid dlci received
                on this lport, to be used in troubleshooting faulty
                configurations."
        ::={ lportEntry 175 }

lportTrkProtState OBJECT-TYPE
        SYNTAX  INTEGER {
                    enabled (1),
                    disabled (2),
                    none (3)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The current operational state of the link trunk protocol
                on this link."
        ::={ lportEntry 176 }

lportTrkTrafficMix OBJECT-TYPE
        SYNTAX  INTEGER {
                    normal (1),
                    management-only (2),
                    management-and-PVCs (3),
                    private (4)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The types of traffic allowed over a Cascade trunk."
        ::= { lportEntry 177 }

lportNumVC OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of VCs going to or through a particular port."
        ::={ lportEntry 178 }

lportTrkAdminCost OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The administrative cost of the trunk. Used when routing
                circuits. Trunks with lower costs are preferred. Value
                ranges from 1 to 65,535."
        ::= { lportEntry 179 }

lportPrivateNet OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "If non-zero, indicates the private network that the
                lport belongs to. If 0, the lport is publicly shared."
        ::= { lportEntry 180 }

lportTrkStaticDelay OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The measured round-trip delay of the trunk, in
                milliseconds. Measured when the trunk last became
                operational."
        ::= { lportEntry 181 }
```

```
lportTrkDynamicDelay OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The recently measured round-tip delay of the trunk. May
                vary from lportTrkStaticDelay due to congestion, or
                reprovisioning of the underlying media."
        ::= { lportEntry 182 }

lportAtmDataRateQoS1 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "An estimate of the logical port's data rate in bits per
                second reserved for QoS class 1."
        ::= { lportEntry 183 }

lportAtmDataRateQoS2 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "An estimate of the logical port's data rate in bits per
                second reserved for QoS class 2."
        ::= { lportEntry 184 }

lportAtmDataRateQoS3 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "An estimate of the logical port's data rate in bits per
                second reserved for QoS class 3."
        ::= { lportEntry 185 }

lportAtmDataRateQoS4 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "An estimate of the logical port's data rate in bits per
                second reserved for QoS class 4."
        ::= { lportEntry 186 }

lportOutVAvailbwQoS1 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current outgoing available virtual bandwidth reserved for
                QoS class 1."
        ::= { lportEntry 187 }

lportOutVAvailbwQoS2 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current outgoing available virtual bandwidth reserved for
                QoS class 2."
        ::= { lportEntry 188 }

lportOutVAvailbwQoS3 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current outgoing available virtual bandwidth reserved for
                QoS class 3."
        ::= { lportEntry 189 }

lportOutVAvailbwQoS4 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current outgoing available virtual bandwidth reserved for
                QoS class 4."
        ::= { lportEntry 190 }

lportInVAvailbwQoS1 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current incoming available virtual bandwidth reserved for
                QoS class 1."
        ::= { lportEntry 191 }

lportInVAvailbwQoS2 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current incoming available virtual bandwidth reserved for
                QoS class 2."
        ::= { lportEntry 192 }
```

```
lportInVAvailbwQoS3 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current incoming available virtual bandwidth reserved for
                QoS class 3."
        ::= { lportEntry 193 }

lportInVAvailbwQoS4 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Currently incoming available virtual bandwidth reserved
                for QoS class 4."
        ::= { lportEntry 194 }

lportOutAllocbwQoS1 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current outgoing allocated bandwidth for QoS Class 1."
        ::= { lportEntry 195 }


lportOutAllocbwQoS2 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current outgoing allocated bandwidth for QoS Class 2."
        ::= { lportEntry 196 }

lportOutAllocbwQoS3 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current outgoing allocated bandwidth for QoS Class 3."
        ::= { lportEntry 197 }

lportOutAllocbwQoS4 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current outgoing allocated bandwidth for QoS Class 4."
        ::= { lportEntry 198 }

lportInAllocbwQoS1 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current incoming allocated bandwidth for QoS Class 1."
        ::= { lportEntry 199 }

lportInAllocbwQoS2 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current incoming allocated bandwidth for QoS Class 2."
        ::= { lportEntry 200 }

lportInAllocbwQoS3 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current incoming allocated bandwidth for QoS Class 3."
        ::= { lportEntry 201 }

lportInAllocbwQoS4 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current incoming allocated bandwidth for QoS Class 4."
        ::= { lportEntry 202 }

lportDynamicQoSbw OBJECT-TYPE
        SYNTAX  INTEGER {
                qos-class-1(1),
                qos-class-2(2),
                qos-class-3(4),
                qos-class-4(8)
        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Bitmap defining which QoS Classes should have bandwidth
                allocated dynamically from lportDataRate instead of
                reserving a percentage up-front."
        ::= { lportEntry 203 }
```

```
lportSvcHoldDownTimer OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                    "The duration in seconds (1..255) the network is
                    allowed to re-establish a SVC after network failure
                    before clearing the SVC.  A value of (0) indicates
                    continuous retries by the network until the user
                    clears the SVC."
        ::= { lportEntry 204 }

lportAtmConnectionType OBJECT-TYPE
        SYNTAX  INTEGER {
                private (1),              -- connection does not involve
                                             public network
                public-switch (2),        -- connection between private
                                             and public networks
                public-endsystem (3)      -- connection between public
                                             network and endsystem
        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION "Type of network connection at this logical port."
        ::= { lportEntry 205 }

lportAtmRouteMetricQoS1 OBJECT-TYPE
        SYNTAX  INTEGER {
                    administrative-cost (1),
                    end-to-end-delay (2),
                    cell-delay-variation (3)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Routing metric associated with QoS Class 1."
        ::= { lportEntry 206 }

lportAtmRouteMetricQoS2 OBJECT-TYPE
        SYNTAX  INTEGER {
                    administrative-cost (1),
                    end-to-end-delay (2),
                    cell-delay-variation (3)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION "Routing metric associated with QoS Class 2."
        ::= { lportEntry 207 }

lportAtmRouteMetricQoS3 OBJECT-TYPE
        SYNTAX  INTEGER {
                    administrative-cost (1),
                    end-to-end-delay (2),
                    cell-delay-variation (3)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION "Routing metric associated with QoS Class 3."
        ::= { lportEntry 208 }

lportAtmRouteMetricQoS4 OBJECT-TYPE
        SYNTAX  INTEGER {
                    administrative-cost (1),
                    end-to-end-delay (2),
                    cell-delay-variation (3)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Routing metric associated with QoS Class 4."
        ::= { lportEntry 209 }

lportIlmiPollTimeout OBJECT-TYPE
        SYNTAX  INTEGER (1..255)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "If ILMI is enabled for this ATM port, the duration in
                seconds that the ILMI poll is generated, if DCE, or
                monitored, if DTE. The default
                value is 5 seconds."
        ::= { lportEntry 210 }

lportAtmProtocol OBJECT-TYPE
        SYNTAX INTEGER {
                uni-30 (1),
                uni-31 (2),
                iisp (3),
                bici-11 (4)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "ATM protocol and its version supported at this ATM port."
        ::= { lportEntry 211 }
```

```
lportIlmiAdminStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                    enabled (1),
                    disabled (2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Administrative state for ILMI function on this ATM port.
                When enabled on DCE ports, the port will actively transmit
                polls and monitor responses in order to determine the
                operational status of the port.  When enabled for DTE
                ports, the port will passively monitor polls to determine
                the operational status of the port."
        ::= { lportEntry 212 }

lportIlmiOperStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                    down (1),
                    registering (2),
                    up (3)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Operational status for ILMI function on this ATM port."
        ::= { lportEntry 213 }

lportIlmiPollRetry OBJECT-TYPE
        SYNTAX  INTEGER (1..255)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "If ILMI is enabled for this ATM port, the consecutive
                missed poll threshold to be reached before declaring the
                port's operational status down. The default value is 4
                times."
        ::= { lportEntry 214 }

lportAtmVpiBits OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of bits of VPI supported."
        ::= { lportEntry 215 }

lportAtmVciBits OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of bits of VCI supported."
        ::= { lportEntry 216 }

lportAtmOamAlarmEnable OBJECT-TYPE
        SYNTAX  INTEGER {
                enabled(1),
                disabled(2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "If disabled then OAM alarms are not generated for
                circuits that are down on this lport."
        ::= { lportEntry 217 }

lportInVAvailbw OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Currently available Incoming virtual bandwidth."
        ::= { lportEntry 218 }

lportbwUNIPolicy OBJECT-TYPE
        SYNTAX  INTEGER {
                enable-uni-bw-policing(1),
                disable-uni-bw-policing(2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The network policy towards requiring sufficient bandwidth
                at the UNI interface prior to allowing a circuit to be
                established."
        ::= { lportEntry 219 }

lportStarvation OBJECT-TYPE
        SYNTAX  INTEGER {
                ok(1),
                error(2)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
               "The CE/CBR card detected a starvation condition on this
                lport for the extended period defined in Bellcore TA-NWT-
                001248"
        ::= { lportEntry 220 }
```

```
lportRecOverflow OBJECT-TYPE
        SYNTAX  INTEGER {
                ok(1),
                error(2)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
               "The CE/CBR card detected a receive overflow condition on
                this lport for the extended period defined in Bellcore TA-
                NWT-001248"
        ::= { lportEntry 221 }

lportLossOfCellSequence OBJECT-TYPE
        SYNTAX  INTEGER {
                ok(1),
                error(2)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
               "The CE/CBR card detected a LOS condition on this lport
                for the extended period defined in Bellcore TA-NWT-001248"
        ::= { lportEntry 222 }

lportLossOfStructurePointer OBJECT-TYPE
        SYNTAX  INTEGER {
                ok(1),
                error(2)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
               "The structured CBR card detected a loss of structure
                pointer condition on this lport
                for the extended period defined in Bellcore TA-NWT-001248"
        ::= { lportEntry 223 }

lportCbrInsDummyCells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of dummy cells inserted due to cell loss on a
                constant bit rate (AAL1) interface."
        ::= { lportEntry 224 }

lportRecFifoUnderflowCnt OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of receive FIFO underflows since the last
                reset. If this condition persists, a lportCBRLineDataError
                trap is issued indicating the lportStarvation."
        ::= { lportEntry 225 }

lportRecFifoOverflowCnt OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of receive FIFO overflows since the last reset.
                If this condition persists, a lportCBRLineDataError trap
                is issued indicating the receive FIFO overflow."
        ::= { lportEntry 226 }

lportCbrLossOfStructurePointerCnt OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The object counts how often the structured CBR card
                detected a loss of the structure pointer since the last
                reset."
        ::= { lportEntry 227 }

lportCbrLossOfCellSequenceCnt OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of loss of cell sequence since the last reset."
        ::= { lportEntry 228 }

lportShaperId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The shaper to be used for this ATM trunk interworking
                with frame relay:

                1-port ATM-IWU STM-1/STS-3c card
                --------------------------------
                Values: 1..16"
        ::= { lportEntry 229 }
```

```
lportDteIlmiPrefixScreenMode OBJECT-TYPE
        SYNTAX  INTEGER {
                    node-prefix           (1),
                    port-prefix           (2),
                    node-prefix-or-port-prefix (3),
                    reject-all            (127),
                    accept-all            (255)
        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The type of screening, if any, to apply against dynamic
                prefixes received from the peer ILMI entity at this ATM
                DTE port."
        ::= { lportEntry 230 }

lportSmdsPduViolTca OBJECT-TYPE
        SYNTAX  INTEGER {
                disabled (1),
                enabled  (2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "This object defines the ability to enable and disable SMDS
                pdu violation traps on this logical port. This
                functionality applies to the following logical port types:
                SMDS DXI/SNI DCE
                SMDS DXI/SNI DTE
                SMDS SSI DTE
                SMDS Optimum Trunk
                Direct Line Trunk
                The default value of this object is 'disabled'."
        ::= {lportEntry 231 }

lportSmdsPduViolThresh OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "This object defines the Smds pdu violation threshold for
                this logical port. The allowable range of threshold values
                is 1-255.  The default value of this object is '10'."
        ::= { lportEntry 232 }

lportSmdsPduHdrSip3SaNotFound OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This object provides the Smds destination address and Smds
                source address of the last Smds pdu to cause the violation
                Smds Sa Not Found. The first 8 bytes of the octet string
                (i.e.; 16 digits in BCD format) correspond to the
                destination address. The second 8 bytes correspond to the
                source address."
        ::= { lportEntry 233 }

lportSmdsPduHdrSip3SaDaOnSamePort OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This object provides the Smds destination address and Smds
                source address of the last Smds pdu to cause the violation
                Smds Sa Da On Same Port. The first 8 bytes of the octet
                string (i.e.; 16 digits in BCD format) correspond to the
                destination address. The second 8 bytes correspond to the
                source address."
        ::= { lportEntry 234 }

lportSmdsPduHdrSip3DstGaNotFound OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This object provides the Smds destination address and Smds
                source address of the last Smds pdu to cause the violation
                Smds Dst Ga Not Found. The first 8 bytes of the octet
                string (i.e.; 16 digits in BCD format) correspond to the
                destination address. The second 8 bytes correspond to the
                source address."
        ::= { lportEntry 235 }

lportSmdsPduHdrSip3DstIaScrnFail OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This object provides the Smds destination address and Smds
                source address of the last Smds pdu to cause the violation
                Smds Dst Ia Scrn Fail. The first 8 bytes of the octet
                string (i.e.; 16 digits in BCD format) correspond to the
                destination address. The second 8 bytes correspond to the
                source address."
        ::= { lportEntry 236 }
```

```
lportSmdsPduHdrSip3SaValFail OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This object provides the Smds destination address and Smds
                source address of the last Smds pdu to cause the violation
                Smds Sa Val Fail. The first 8 bytes of the octet string
                (i.e.; 16 digits in BCD format) correspond to the
                destination address. The second 8 bytes correspond to the
                source address."
        ::= { lportEntry 237 }

lportSmdsPduHdrSip3DstIaNotFound OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This object provides the Smds destination address and Smds
                source address of the last Smds pdu to cause the violation
                Smds Dst Ia Not Found. The first 8 bytes of the octet
                string (i.e.; 16 digits in BCD format) correspond to the
                destination address. The second 8 bytes correspond to the
                source address."
        ::= { lportEntry 238 }

lportSmdsPduHdrSip3SrcIaScrnFail OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This object provides the Smds destination address and Smds
                source address of the last Smds pdu to cause the violation
                Smds Src Ia Scrn Fail. The first 8 bytes of the octet
                string (i.e.; 16 digits in BCD format) correspond to the
                destination address. The second 8 bytes correspond to the
                source address."
        ::= { lportEntry 239 }

lportSmdsPduHdrSip3DstGaScrnFail OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This object provides the Smds destination address and Smds
                source address of the last Smds pdu to cause the violation
                Smds Dst Ga Scrn Fail. The first 8 bytes of the octet
                string (i.e.; 16 digits in BCD format) correspond to the
                destination address. The second 8 bytes correspond to the
                source address."
        ::= { lportEntry 240 }

lportSmdsPduHdrSip3SaTypeInvalid OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This object provides the Smds destination address and Smds
                source address of the last Smds pdu to cause the violation
                Smds Sa Type Invalid. The first 8 bytes of the octet string
                (i.e.; 16 digits in BCD format) correspond to the
                destination address. The second 8 bytes correspond to the
                source address."
        ::= { lportEntry 241 }

lportSmdsPduHdrSip3DaTypeInvalid OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This object provides the Smds destination address and Smds
                source address of the last Smds pdu to cause the violation
                Smds Da Type Invalid. The first 8 bytes of the octet string
                (i.e.; 16 digits in BCD format) correspond to the
                destination address. The second 8 bytes correspond to the
                source address."
        ::= { lportEntry 242 }

lportSmdsPduHdrDxi2LinkIdInvalid OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This object provides the dxi2 header of the last Smds pdu
                to cause the violation Dxi2 Link Id Invalid. The length of
                this object is 4 bytes."
        ::= { lportEntry 243 }

lportSmdsPduHdrDxi2CrInvalid OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This object provides the dxi2 header of the last Smds pdu
                to cause the violation Dxi2 Cr Invalid. The length of this
                object is 4 bytes."
        ::= { lportEntry 244 }
```

```
lportSmdsPduHdrDxi2CtrlInvalid OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This object provides the dxi2 header of the last Smds pdu
                to cause the violation Dxi2 Ctrl Invalid. The length of
                this object is 4 bytes."
        ::= { lportEntry 245 }

lportSmdsPduHdrDxi2StationIdInvalid OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This object provides the dxi2 header of the last Smds pdu
                to cause the violation Dxi2 Station Id Invalid. The length
                of this object is 4 bytes."
        ::= { lportEntry 246 }

lportSmdsPduHdrDxi2AeInvalid OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This object provides the dxi2 header of the last Smds pdu
                to cause the violation Dxi2 Ae Invalid. The length of this
                object is 4 bytes."
        ::= { lportEntry 247 }

lportDS0FarendLpbkEnabled OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Set a single DS0 into farend loopback 1-24."
        ::= { lportEntry 248 }

lportDS0FarendLpbkDisabled OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Set a single DS0 out farend loopback 1-24."
        ::= { lportEntry 249 }

lportTrkProtFailureThresholdOBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The time, measured in seconds, that Trunk Protocol waits
                before transitioning the state of an unresponsive trunk
                from UP to DOWN.  Trunk protocol keepalive requests are
                issued on the trunk lport once per second.  This value
                defines the failure threshold, that is, the number of
                consecutive requests that must go unanswered before the
                Trunk Protocol application will declare the trunk lport
                DOWN."
        ::= { lportEntry 250 }

lportPtr    OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "An octet string indicating the lport pointer."
        ::= { lportEntry 251 }

lportISDNPoolUtil OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The percent utilization of the B-channel pool that
                this b-channel belongs to."
        ::= { lportEntry 252 }

lportISDNIpAddrAssignFail OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The cause value for why Ip Address assignment
                has failed."
        ::= { lportEntry 253 }

lportTrkUtilQoS1 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "An estimate of the real QoS1 utilization of the trunk
                bandwidth in terms of a percentage of the configured trunk
                bandwidth."
        ::= { lportEntry 254 }
```

```
lportTrkUtilQoS2 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "An estimate of the real QoS2 utilization of the trunk
                bandwidth in terms of a percentage of the configured trunk
                bandwidth."
        ::= { lportEntry 255 }

lportTrkUtilQoS3 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "An estimate of the real QoS3 utilization of the trunk
                bandwidth in terms of a percentage of the configured trunk
                bandwidth."
        ::= { lportEntry 256 }

lportTrkUtilQoS4 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "An estimate of the real QoS4 utilization of the trunk
                bandwidth in terms of a percentage of the configured trunk
                bandwidth."
        ::= { lportEntry 257 }

lportIlmiNumOctetsTx OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of ILMI octets transmitted on this ATM port."
        ::= { lportEntry 258 }

lportIlmiNumOctetsRx OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of ILMI octets received on this ATM port."
        ::= { lportEntry 259 }

lportIlmiNumPdusTx OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of ILMI PDU's transmitted on this ATM port."
        ::= { lportEntry 260 }

lportIlmiNumPdusRx OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of properly formatted ILMI PDU's received on
                this ATM port."
        ::= { lportEntry 261 }

lportIlmiNumErrorsRx OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of improperly formatted ILMI PDU's received on
                this ATM port."
        ::= { lportEntry 262 }

lportIlmiNumUmePollsTx OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of ILMI polls sent by the UME entity on this
                ATM DCE port."
        ::= { lportEntry 263 }

lportIlmiNumUmeResponsesRx OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of ILMI poll responses received by the UME
                entity on this ATM DCE port."
        ::= { lportEntry 264 }

lportIlmiVPI OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The VPI value of the VCC provisioned for the ILMI. The
                default value is 0."
        ::= { lportEntry 265 }

lportIlmiVCI OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The VCI value of the VCC provisioned for the ILMI. The
                default value is 16."
        ::= { lportEntry 266 }
```

```
lportInCells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of cells received"
        ::= { lportEntry 267 }

lportOutCells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of cells transmitted"
        ::= { lportEntry 268 }

lportDS1ChannelId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The channel id of the DS1 channel for the channelized DS3
                IOP."
        ::= { lportEntry 269 }

lportCDV OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The Cell Delay Variation of this port measured in
                microseconds."
        ::= { lportEntry 270 }

lportAtmTrkIomCktDiagStr OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Internal diagnostic information."
        ::= { lportEntry 271 }

lportAtmTrkSpCktDiagStr OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Internal diagnostic information."
        ::= { lportEntry 272 }

lportAuthState OBJECT-TYPE
        SYNTAX  INTEGER {
                auth-enabled (1),
                auth_disabled (2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Authentication enabled for this port, yes or no."
        ::= { lportEntry 273 }

lportAuthDomainID OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Authentication Domain ID for this lport."
        ::= { lportEntry 274 }

lportAuthPPPOption OBJECT-TYPE
        SYNTAX  INTEGER {
                pap-only (1),
                chap-only (2),
                pap-and-chap (3)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "PPP authentication options."
        ::= { lportEntry 275 }
```

```
lportAuthFailReason OBJECT-TYPE
        SYNTAX  INTEGER {
                syserNoDomain   (1),
                syserNoServer   (2),
                syserInvAuthMethod (3),
                syserNoUsername  (4),
                syserNoPassword  (5),
                syserNoSecret   (6),
                syserNoPlChallenge (7),
                syserNoEnChallenge (8),
                syserNoPortId   (9),
                syserSendError (10),
                noRespFromServer(11),
                invRespFromServer(12),
                invServiceType (13),
                invFramedProtocol(14),
                invDomainId    (15),
                authenticationFailed(16)
        }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Reason for authentication failure - trap sent to NMS."
        ::= { lportEntry 276 }

lportEchoRequestOption OBJECT-TYPE
        SYNTAX  INTEGER {
                echo_rqst_enabled  (1),
                echo_rqst_disabled (2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The sending of Echo Requests enabled for this port, yes
                or no."
        ::= { lportEntry 277 }

lportEchoRequestInterval OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The # seconds to wait between sending Echo Requests."
        ::= { lportEntry 278 }

lportEchoRequestMaxTries OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The max # times to send an Echo Request without getting a
                 response."
        ::= { lportEntry 279 }

lportMultilinkProtocolOption  OBJECT-TYPE
        SYNTAX  INTEGER {
                mp_enabled  (1),
                mp_disabled (2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "PPP Multilink Protocol enabled for this port, yes or no."
        ::= { lportEntry 280 }

lportMultilinkProtocolFailReason OBJECT-TYPE
        SYNTAX  INTEGER {
                failBundleCreate (1),
                failLinkAdd (2)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Failure to create an MP bundle or to add a link to an
                 MP Bundle - trap sent."
        ::= { lportEntry 281 }

lportBandwidthAllocProtocolOption  OBJECT-TYPE
        SYNTAX  INTEGER {
                bap_enabled  (1),
                bap_disabled (2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "PPP Bandwidth Alloc Protocol enabled for this port, yes
                or no."
        ::= { lportEntry 282 }

lportBandwidthAllocProtocolCallFailReason OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "BAP Call Fail Status code (Q.931 cause code) - trap sent
                 to NMS."
        ::= { lportEntry 283 }
```

```
lportPrivateNetOverflow OBJECT-TYPE
        SYNTAX  INTEGER {
                    restrict (0),
                    use-public (1)
                }
       ACCESS  read-write
       STATUS  mandatory
       DESCRIPTION
              "Indicates how circuits belonging to private lports
              are handled, when the resources of the network have become
              exhausted. If set to use-public, the resources of the
              public network can be used during overflow conditions."
       ::= { lportEntry 284 }

lportCbrFifoHalfLength OBJECT-TYPE
       SYNTAX INTEGER (2..8)
       ACCESS read-write
       STATUS mandatory
       DESCRIPTION
              "Determines the size of the FIFO used on the CBR COBRA.
              The unit are the number of cells stored in the FIFO in
              direction towards the line interface.
              The size of the FIFO influences the average cell delay
              and the maximum tolerable cell delay variation.
              In ACM mode (pportCbrCurrentClockMode=3) the COBRA tries
              to keep the FIFO filled up to lportCbrFifoHalfLength and
              adjusts the line speed accordingly.
              In all other clock modes (STRS, synchronous) the line speed
              cannot be varied.
              Decreasing this value from its default (and maximum) of 8
              reduces the cell (and ultimately the line) delay at the
              cost of decreasing the cell delay variation that can be
              handled."
       ::= { lportEntry 285 }

lportCustomerID OBJECT-TYPE
       SYNTAX INTEGER
       ACCESS read-write
       STATUS mandatory
       DESCRIPTION
              "The Customer that owns this lport. For Virtual
              Private Networking Support."
       ::= { lportEntry 286 }

lportCongestThresh0 OBJECT-TYPE
       SYNTAX INTEGER
       ACCESS read-write
       STATUS mandatory
       DESCRIPTION
              "Congestion threshold #0 for NTM/NDC in cells/second.
              Used as a severe congestion abatement threshold on an IOM."
       ::= { lportEntry 287 }

lportCongestThresh1 OBJECT-TYPE
       SYNTAX INTEGER
       ACCESS read-write
       STATUS mandatory
       DESCRIPTION
              "Congestion threshold #1 for NTM/NDC in cells/second.
              Used as a low congestion threshold on an IOM."
       ::= { lportEntry 288 }

lportCongestThresh2 OBJECT-TYPE
       SYNTAX INTEGER
       ACCESS read-write
       STATUS mandatory
       DESCRIPTION
              "Congestion threshold #2 for NTM/NDC in cells/second
              Used as a high congestion threshold on an IOM."
       ::= { lportEntry 289 }

lportCongestThresh3 OBJECT-TYPE
       SYNTAX INTEGER
       ACCESS read-write
       STATUS mandatory
       DESCRIPTION
              "Congestion threshold #3 for NTM/NDC in cells/second.
              Used as a severe congestion threshold on an IOM."
       ::= { lportEntry 290 }

lportSevereCongestNotifyTime OBJECT-TYPE
       SYNTAX INTEGER
       ACCESS read-write
       STATUS mandatory
       DESCRIPTION
              "Selects a minimum severe congestion period upon which
              an alarm is generated on an IOM. Default value is
              30 seconds."
       ::= { lportEntry 291 }

lportSevereCongestStatus OBJECT-TYPE
       SYNTAX INTEGER {
              notCongested (1),
              congested (2)
       }
       ACCESS read-only
       STATUS mandatory
       DESCRIPTION
              "Indicates the state of severe congestion on a logical
              port."
       ::= { lportEntry 292 }
```

```
lportSmdsNumInFramesIa OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Number of SMDS IA Frames received on an SSI/DXI-SNI
                lport."
        ::= { lportEntry 293 }

lportSmdsNumInBytesIa OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Number of SMDS IA Bytes received on an SSI/DXI-SNI
                lport."
        ::= { lportEntry 294 }

lportSmdsNumInFramesGa OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Number of SMDS GA Frames received on an SSI/DXI-SNI
                lport."
        ::= { lportEntry 295 }

lportSmdsNumInBytesGa OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Number of SMDS GA Bytes received on an SSI/DXI-SNI
                lport."
        ::= { lportEntry 296 }

lportSmdsNumOutFramesIa OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Number of SMDS IA Frames transmitted on an SSI/DXI-SNI
                lport."
        ::= { lportEntry 297 }

lportSmdsNumOutBytesIa OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Number of SMDS IA Bytes transmitted on an SSI/DXI-SNI
                lport."
        ::= { lportEntry 298 }

lportSmdsNumOutFramesGa OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Number of SMDS GA Frames transmitted on an SSI/DXI-SNI
                lport."
        ::= { lportEntry 299 }

lportSmdsNumOutBytesGa OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Number of SMDS GA Bytes transmitted on an SSI/DXI-SNI
                lport."
        ::= { lportEntry 300 }

lportBadPVCFactor OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
        "The factor used to determine the threshold for bad PVC
        detection. Its value ranges from 0 to 32. The relationship
        between the threshold and the factor is defined as:
        Bc + ( Be / 2 )
        Threshould = ---------------------- ,
        2 ^ (32-Fb)
        where Fb is the factor. By default, it is set to 30. "
        ::= {lportEntry 305 }

lportAmberReductionPm OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
        "The percentage of amber frame reduction when mild congestion
        happens, by default, it is set to 50."
        ::= {lportEntry 306 }

lportAmberReductionPs OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
        "The percentage of amber frame reduction when severe congestion
        happens, by default, it is set to 75."
        ::= {lportEntry 307 }
```

```
lportCongestionCheckInterval OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Specifies,
                1. when Congestion Control Switch is set to 'OSPF based
                close loop control', the interval between successive
                congestion state checking at the port and in this
                case it must be a multiple of 1 second (default is 1
                second),
                2. when Congestion Control Switch is set to 'Per-VC based
                close loop control', the checking interval before sending
                a high priority congestion control notification
                packet after sending the first normal congestion control
                notification packet and failed to get any improvement on
                the port congestion state, and in this
                case it must be a multiple of 1 milli-second (default is
                1000 milli-secondss). "
        ::= {lportEntry 308 }

lportCongestionClearDelay OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Specifies,
                1. when Congestion Control Switch is set to 'OSPF based
                close loop control', the delay before congestion clearance
                message is sent when the congested port
                becomes less congested, and in this case it must be a
                multiple of 1 second (default is 3 seconds),
                2. when Congestion Control Switch is set to 'Per-VC based
                close loop control', the delay for clearing the congestion
                reaction at the ingress switch port after
                receiving an less congestion notification message, in this
                case it must be a multiple of 1 milli-second. (default is
                3000 milli-seconds)"
        ::= {lportEntry 309 }

lportNrtsRmGenType OBJECT-TYPE
       SYNTAX INTEGER {
                none (1),
                ccrm (2),
                bcm (3)
       }
       ACCESS read-write
       STATUS mandatory
       DESCRIPTION
                "The type of RM cells the NRTS processor generates on this
                lport."
       ::= { lportEntry 310 }

lportNrtsRmTermType OBJECT-TYPE
       SYNTAX INTEGER {
                ccrmOnly (1),
                ccrmAndBcm (2)
       }
       ACCESS read-write
       STATUS mandatory
       DESCRIPTION
                "The type of RM cells the NRTS processor terminates on this
                lport."
       ::= { lportEntry 311 }

lportNrtsEfciCheck OBJECT-TYPE
       SYNTAX INTEGER {
                no (1),
                yes (2)
       }
       ACCESS read-write
       STATUS mandatory
       DESCRIPTION
                "Whether the NRTS processor checks the EFCI bit for
                circuits on this lport when incrementing EFCI counts."
       ::= { lportEntry 312 }

lportNrtsBufAlloc OBJECT-TYPE
       SYNTAX INTEGER
       ACCESS read-write
       STATUS mandatory
       DESCRIPTION
                "Cell buffer size for this lport controlled by the NRTS
                processor."
       ::= { lportEntry 313 }

lportNrtsClp01Thresh OBJECT-TYPE
       SYNTAX INTEGER
       ACCESS read-write
       STATUS mandatory
       DESCRIPTION
                "The CLP=0+1 threshold for this lport used by the NRTS
                processor. Must be smaller than the cell buffer size
                allocated for this lport."
       ::= { lportEntry 314 }

lportNrtsDiscardThresh OBJECT-TYPE
       SYNTAX INTEGER
       ACCESS read-write
       STATUS mandatory
       DESCRIPTION
                "The CLP=1 discard threshold for this lport used by the
                NRTS processor. Must be smaller than the CLP=0+1
                threshold. May also be used as the EPD threshold."
       ::= { lportEntry 315 }
```

```
lportNrtsEfciThresh OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The EFCI threshold for this lport used by the NRTS
                processor. Must be smaller than the CLP=0+1 and discard
                thresholds."
        ::= { lportEntry 316 }

lportNrtsRmCellCount OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The number of RM cells received on this lport by the NRTS
                processor."
        ::= { lportEntry 317 }

lportCloseLoopSwitch OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Close loop congestion control function switch on this
                logical port.
                0:   Close loop control OFF,
                1:   OSPF based close loop congestion control,
                2:   Per-VC based close loop congestion control.
                By default, it is set to 0. "
        ::= {lportEntry 318 }

lportAtmVPIStop OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The stop value for the PVC VPI range.
                Used only by VP Mux termination lports"
        ::= {lportEntry 319 }

lportServiceClassType OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Specifies the service class types to be supported by the
                logical port.
                the valid values and their corresponding definitions are:
                0: mono-class,
                1: multi-class"
        ::= {lportEntry 320 }

lportSegmentation OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Specifies the transportation feature of a trunk port. If
                enabled, the port cuts user packets
                into fix-length segments for delivery over the trunk,
                otherwise user packets are delivered over
                the trunk without segmentation. If the port service class
                type is configured as 'muti-class',
                the default is 'enabled'. If the port service class type
                is configured as 'mono-class', the default is 'disabled'."
        ::= {lportEntry 321 }

lportTrkOSPFAreaID OBJECT-TYPE
        SYNTAX IpAddress
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "OSPF Area ID to which the trunk belongs. Default
                value is 0.0.0.1 for historical reasons."
        ::= { lportEntry 322 }

lportMultiServiceCBR OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Specifies that CBR class is to be supported on the
                logical port."
        ::= { lportEntry 323 }

lportMultiServiceVBRrt OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
        "Specifies that VBR_rt class is to be supported on the logical
        port."
        ::= { lportEntry 324 }

lportMultiServiceVBRnrt OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
        "Specifies that VBR_nrt class is to be supported on the logical
        port."
        ::= { lportEntry 325 }
```

```
lportMultiServiceUBR OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Specifies that UBR class is to be supported on the
                logical port."
        ::= { lportEntry 326 }
```

The Network Traffic Management table.

```
-- This table contains NTM statistics per GR-1248.
--
lportNtmTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF LportNtmEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A list of Network Traffic Management statistics for
                a logical port on an IOM."
        ::= { lport 3 }

lportNtmEntry OBJECT-TYPE
        SYNTAX  LportNtmEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A record of Network Traffic Management statistics
                for a logical port on an IOM."
        INDEX { lportNtmIfIndex, lportNtmHistIndex }
        ::= { lportNtmTable 1 }

LportNtmEntry ::=
        SEQUENCE {
                lportNtmIfIndex
                    Index,
                lportNtmHistIndex
                    INTEGER,
                lportNtmTimeStamp
                    INTEGER,
                lportNtmOutDiscardCells
                    INTEGER,
                lportNtmInMcl0
                    INTEGER,
                lportNtmEnterMcl0
                    INTEGER,
                lportNtmInMcl1
                    INTEGER,
                lportNtmEnterMcl1
                    INTEGER,
                lportNtmInMcl2
                    INTEGER,
                lportNtmEnterMcl2
                    INTEGER,
                lportNtmInMcl3
                    INTEGER,
                lportNtmEnterMcl3
                    INTEGER
                }

lportNtmIfIndex OBJECT-TYPE
        SYNTAX Index
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The ifIndex value of the  corresponding ifEntry."
        ::= { lportNtmEntry 1 }

lportNtmHistIndex OBJECT-TYPE
        SYNTAX  INTEGER {
                current (1),
                history1 (2),
                history2 (3),
                history3 (4)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "An index of the Network Traffic Management history."
        ::= { lportNtmEntry 2 }

lportNtmTimeStamp OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "For the current counts (lportNTMhistIndex of 1),
                time elapsed in the current 5-minute NTM collection
                interval. For the history counts (lportNTMhistIndex
                of 2 to 4), timestamp at the end of 5-minute NTM
                collection interval. Resolution is 1 second."
        ::= { lportNtmEntry 3 }

lportNtmOutDiscardCells OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A count of the outgoing CLP=0+1 user and OAM cells
                discarded within the Network Traffic Management
                5-minute interval."
        ::= { lportNtmEntry 4 }
```

```
lportNtmInMcl0 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A number of times the smoothed Measure of Congestion
                is in Machine Congestion Level #0. Counts the number
                of 20 ms periods within the Network Traffic Management
                5-minute interval."
        ::= { lportNtmEntry 5 }

lportNtmEnterMcl0 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A number of times the smoothed Measure of Congestion
                enters Machine Congestion Level #0. Counts the number of
                20 ms periods within the Network Traffic Management
                5-minute collection interval."
        ::= { lportNtmEntry 6 }

lportNtmInMcl1 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A number of times the smoothed Measure of Congestion
                is in Machine Congestion Level #1. Counts the number
                of 20 ms periods within the Network Traffic Management
                5-minute interval."
        ::= { lportNtmEntry 7 }

lportNtmEnterMcl1 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A number of times the smoothed Measure of Congestion
                enters Machine Congestion Level #1. Counts the number of
                20 ms periods within the Network Traffic Management
                5-minute collection interval."
        ::= { lportNtmEntry 8 }

lportNtmInMcl2 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A number of times the smoothed Measure of Congestion
                is in Machine Congestion Level #2. Counts the number
                of 20 ms periods within the Network Traffic Management
                5-minute interval."
        ::= { lportNtmEntry 9 }

lportNtmEnterMcl2 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A number of times the smoothed Measure of Congestion
                enters Machine Congestion Level #2. Counts the number of
                20 ms periods within the Network Traffic Management
                5-minute collection interval."
        ::= { lportNtmEntry 10 }

lportNtmInMcl3 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A number of times the smoothed Measure of Congestion
                is in Machine Congestion Level #3. Counts the number
                of 20 ms periods within the Network Traffic Management
                5-minute interval."
        ::= { lportNtmEntry 11 }

lportNtmEnterMcl3 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A number of times the smoothed Measure of Congestion
                enters Machine Congestion Level #3. Counts the number of
                20 ms periods within the Network Traffic Management
                5-minute collection interval."
        ::= {lportNtmEntry 12 }
```

The Network Data Collection table.

```
-- This table contains NDC statistics per GR-1248.
lportNdcTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF LportNdcEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A list of Network Data Collection statistics for
                a logical port on an IOM."
        ::= { lport 4 }
```

```
lportNdcEntry OBJECT-TYPE
        SYNTAX  LportNdcEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A record of Network Data Collection statistics for
                a logical port on an IOM."
        INDEX { lportNdcIfIndex, lportNdcHistIndex }
        ::= { lportNdcTable 1 }

LportNdcEntry ::=
        SEQUENCE {
                lportNdcIfIndex
                    Index,
                lportNdcHistIndex
                    INTEGER,
                lportNdcTimeStamp
                    INTEGER,
                lportNdcInCells
                    INTEGER,
                lportNdcOutCells
                    INTEGER,
                lportNdcOutDiscardCells
                    INTEGER,
                lportNdcInMcl0
                    INTEGER,
                lportNdcEnterMcl0
                    INTEGER,
                lportNdcInMcl1
                    INTEGER,
                lportNdcEnterMcl1
                    INTEGER,
                lportNdcInMcl2
                    INTEGER,
                lportNdcEnterMcl2
                    INTEGER,
                lportNdcInMcl3
                    INTEGER,
                lportNdcEnterMcl3
                    INTEGER
                }

lportNdcIfIndex OBJECT-TYPE
        SYNTAX Index
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The ifIndex value of the  corresponding ifEntry."
        ::= { lportNdcEntry 1 }

lportNdcHistIndex OBJECT-TYPE
        SYNTAX  INTEGER {
                current (1),
                history1 (2),
                history2 (3)
        }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "An index of the Network Data Collection history."
        ::= { lportNdcEntry 2 }

lportNdcTimeStamp OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "For the current counts (lportNDChistIndex of 1),
                time elapsed in the current 15-minute NDC collection
                interval. For the history counts (lportNDChistIndex
                of 2 to 3), timestamp at the end of 15-minute NDC
                collection interval. Resolution is 1 second."
        ::= { lportNdcEntry 3 }

lportNdcInCells OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A count of the incoming CLP=0+1 user and OAM cells
                received within the Network Data Collection
                15-minute interval."
        ::= { lportNdcEntry 4 }

lportNdcOutCells OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A count of the outgoing CLP=0+1 user and OAM
                cells transmitted within the Network Data Collection
                15-minute interval."
        ::= { lportNdcEntry 5 }

lportNdcOutDiscardCells OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A count of the outgoing CLP=0+1 user and OAM
                cells discarded within the Network Data Collection
                15-minute interval."
        ::= { lportNdcEntry 6 }
```

```
lportNdcInMcl0 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A number of times the smoothed Measure of Congestion
                is in Machine Congestion Level #0. Counts the number
                of 20 ms periods within the Network Data Collection
                15-minute interval."
        ::= { lportNdcEntry 7 }

lportNdcEnterMcl0 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A number of times the smoothed Measure of Congestion
                enters Machine Congestion Level #0. Counts the number
                of 20 ms periods within the Network Data Collection
                15-minute interval."
        ::= { lportNdcEntry 8 }

lportNdcInMcl1 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A number of times the smoothed Measure of Congestion
                is in Machine Congestion Level #1. Counts the number
                of 20 ms periods within the Network Data Collection
                 15-minute interval."
        ::= { lportNdcEntry 9 }

lportNdcEnterMcl1 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A number of times the smoothed Measure of Congestion
                enters Machine Congestion Level #1. Counts the number
                of 20 ms periods within the Network Data Collection
                15-minute interval."
        ::= { lportNdcEntry 10 }

lportNdcInMcl2 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A number of times the smoothed Measure of Congestion
                is in Machine Congestion Level #2. Counts the number
                of 20 ms periods within the Network Data Collection
                15-minute interval."
        ::= { lportNdcEntry 11 }

lportNdcEnterMcl2 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A number of times the smoothed Measure of Congestion
                enters Machine Congestion Level #2. Counts the number
                of 20 ms periods within the Network Data Collection
                15-minute interval."
        ::= { lportNdcEntry 12 }

lportNdcInMcl3 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A number of times the smoothed Measure of Congestion
                is in Machine Congestion Level #3. Counts the number
                of 20 ms periods within the Network Data Collection
                15-minute interval."
        ::= { lportNdcEntry 13 }

lportNdcEnterMcl3 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A number of times the smoothed Measure of Congestion
                enters Machine Congestion Level #3. Counts the number
                of 20 ms periods within the Network Data Collection
                15-minute interval."
        ::= { lportNdcEntry 14 }
```

The Priority Bandwidth Lport Table

```
-- This table contains the bandwidth consumed on
-- lports by priority level and QoS class for each interface.

lportPriBWTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF LportPriBWEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A list of logical port/bandwidth priority/QoS entries.
                 The number of entries is given by the value of ifNumber
                 in MIB-II."
        ::= { lport 2 }
```

```
lportPriBWEntry OBJECT-TYPE
        SYNTAX  LportPriBWEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "The logical port priority bandwidth entry contains
                 objects relevant to priority bandwidth classes on a
                 logical port."
        INDEX   { lportPriBWIfIndex, lportPriBWClass, lportPriBWQoS }
        ::= { lportPriBWTable 1 }

LportPriBWEntry ::=
        SEQUENCE {
                lportPriBWIfIndex
                    Index,
                lportPriBWClass
                    Index,
                lportPriBWQoS
                    Index,
                lportPriBWNumVC
                    Counter,
                lportPriBWAlloc
                    INTEGER
        }

lportPriBWIfIndex OBJECT-TYPE
        SYNTAX  Index
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The ifIndex value of the  corresponding ifEntry."
        ::= { lportPriBWEntry 1 }

lportPriBWClass OBJECT-TYPE
        SYNTAX  Index
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The priority bandwidth class."
        ::= { lportPriBWEntry 2 }

lportPriBWQoS OBJECT-TYPE
        SYNTAX  Index
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The traffic QoS class."
        ::= { lportPriBWEntry 3 }

lportPriBWNumVC OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of VCs of this interface/class/qos."
        ::= { lportPriBWEntry 4 }

lportPriBWAlloc OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The amount of virtual bandwidth allocated by the
                 VCs of this interface/class/qos."
        ::= { lportPriBWEntry 5 }
```

Lport ATM Signalling Table

```
atmSigLportTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF AtmSigLportEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A table of ATM signalling paramaters associated with
                logical ports. The number of entries is given by the value
                of ifNumber in MIB-II."
        ::= { lport 2 }

atmSigLportEntry OBJECT-TYPE
        SYNTAX  AtmSigLportEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "The ATM Signalling logical port entry contains objects
                relevant to the configuration and monitoring of ATM
                signalling on a logical port."
        INDEX   { atmSigLportIfIndex }
        ::= { atmSigLportTable 1 }

AtmSigLportEntry ::=
        SEQUENCE {
                atmSigLportIfIndex
                    Index,
                atmSigLportAdminStatus
                    INTEGER,
                atmSigLportQ93bMaxRestart
                    INTEGER,
                atmSigLportQ93bMaxStatEnq
                    INTEGER,
                atmSigLportQ93bT303
                    INTEGER,
                atmSigLportQ93bT308
                    INTEGER,
```

```
atmSigLportQ93bT309
    INTEGER,
atmSigLportQ93bT310
    INTEGER,
atmSigLportQ93bT313
    INTEGER,
atmSigLportQ93bT316
    INTEGER,
atmSigLportQ93bT322
    INTEGER,
atmSigLportQ93bT398
    INTEGER,
atmSigLportQ93bT399
    INTEGER,
atmSigLportOperStatus
    INTEGER,
atmSigLportQ93bTotalConns
    Counter,
atmSigLportQ93bActiveConns
    Counter,
atmSigLportQ93bNumPduTx
    Counter,
atmSigLportQ93bLastCauseTx
    INTEGER,
atmSigLportQ93bLastDiagTx
    INTEGER,
atmSigLportQ93bNumPduRx
    Counter,
atmSigLportQ93bLastCauseRx
    INTEGER,
atmSigLportQ93bLastDiagRx
    INTEGER,
atmSigLportQSaalMaxCC
    INTEGER,
atmSigLportQSaalMaxPD
    INTEGER,
atmSigLportQSaalMaxStat
    INTEGER,
atmSigLportQSaalTPoll
    INTEGER,
atmSigLportQSaalTKeepalive
    INTEGER,
atmSigLportQSaalTNoResponse
    INTEGER,
atmSigLportQSaalTCC
    INTEGER,
atmSigLportQSaalTIdle
    INTEGER,
atmSigLportQSaalNumDiscardTx
    Counter,
atmSigLportQSaalNumErrorTx
    Counter,
atmSigLportQSaalNumPduTx
    Counter,
atmSigLportQSaalNumDiscardRx
    Counter,
atmSigLportQSaalNumErrorRx
    Counter,
atmSigLportQSaalNumPduRx
    Counter
}

atmSigLportIfIndex OBJECT-TYPE
    SYNTAX  Index
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The ifIndex value of the  corresponding ifEntry."
    ::= { atmSigLportEntry 1 }

atmSigLportAdminStatus OBJECT-TYPE
    SYNTAX  INTEGER {
        enabled (1),
        disabled (2)
        }
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The configured state of the ATM signalling function for
        this port."
    ::= { atmSigLportEntry 2 }
```

```
atmSigLportQ93bMaxRestart OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The maximum number of unacknowledged restarts to send
                before declaring a signalling failure."
        ::= { atmSigLportEntry 3 }

atmSigLportQ93bMaxStatEnq OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The maximum number of unacknowledged status enquiries to
                send before issuing a restart."
        ::= { atmSigLportEntry 4 }

atmSigLportQ93bT303 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Protocol Timer T303, specified in milliseconds."
        ::= { atmSigLportEntry 5 }

atmSigLportQ93bT308 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Protocol Timer T308, specified in milliseconds."
        ::= { atmSigLportEntry 6 }

atmSigLportQ93bT309 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Protocol Timer T309, specified in milliseconds."
        ::= { atmSigLportEntry 7 }

atmSigLportQ93bT310 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Protocol Timer T310, specified in milliseconds."
        ::= { atmSigLportEntry 8 }

atmSigLportQ93bT313 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Protocol Timer T313, specified in milliseconds."
        ::= { atmSigLportEntry 9 }

atmSigLportQ93bT316 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Protocol Timer T316, specified in milliseconds."
        ::= { atmSigLportEntry 10 }

atmSigLportQ93bT322 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Protocol Timer T322, specified in milliseconds."
        ::= { atmSigLportEntry 11 }

atmSigLportQ93bT398 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Protocol Timer T398, specified in milliseconds."
        ::= { atmSigLportEntry 12 }

atmSigLportQ93bT399 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Protocol Timer T399, specified in milliseconds."
        ::= { atmSigLportEntry 13 }

atmSigLportOperStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                up (1),
                down (2)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The operational status of the signalling function on this
                port."
        ::= { atmSigLportEntry 14 }
```

```
atmSigLportQ93bTotalConns OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of SVC's on this port."
        ::= { atmSigLportEntry 15 }

atmSigLportQ93bActiveConns OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of active SVC's on this port."
        ::= { atmSigLportEntry 16 }
atmSigLportQ93bNumPduTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of Q93B PDU's transmitted on this port."
        ::= { atmSigLportEntry 17 }

atmSigLportQ93bLastCauseTx OBJECT-TYPE
        SYNTAX  INTEGER {
                unallocNmb (1),         -- unallocated (unassigned) number
                noRoutTransnet (2),     -- no route to transit network
                noRoutDst (3),          -- no route to destination
                vccUnacpt (10),         -- UNI 3.0: VPCI/VCI unacceptable
                normCallClr (16),       -- UNI 3.1: normal call clearing
                usrBusy (17),           -- user busy
                nousrRsp (18),          -- no user response
                callRej (21),           -- call rejected
                nmbChng (22),           -- number changed
                callRejClir (23),       -- user rejects all calls with CLIR
                dstOutOrd (27),         -- destination out of order
                invNmbForm (28),        -- invalid number format
                rspStatEnq (30),        -- response to STATUS ENQUIRY
                normUnspec (31),        -- normal unspecified
                reqVccUnavail (35),     -- requested VPCI/VCI unavailable
                vccFail (36),           -- UNI 3.1: VPCI/VCI assignment
                                        -- failure
                rateUnavail1 (37),      -- UNI 3.1: user cell rate
                                        -- unavailable
                netOutOrd (38),         -- network out of order
                tmpFail (41),           -- Temporary failure
                accInfoDisc (43),       -- access info discarded
                noVccAvail (45),        -- no VPCI/VCI unavailable
                resUnavail (47),        -- resources unavailable,
                                        -- unspecified
                qosUnavail (49),        -- Quality of Service unavailable
                rateUnavail (51),       -- UNI 3.0: user cell rate
                                        -- unavailable
                bCapNotAuth (57),       -- bearer capability not authorized
                bCapUnavail (58),       -- bearer capability not available
                srvUnavail (63),        -- Service or option unavailable
                bCapNotImpl (65),       -- bearer capability not implemented
                combUnsupp (73),        -- unsupported comb. of traffic
                                        -- parameters
                aalParmUnsupp1 (78),    -- UNI 3.1: AAL paramteres cannot
                                        -- be supported
                invCallRef (81),        -- invalid call reference
                chanNotExst (82),       -- identified channel does not exist
                dstNotComp (88),        -- incompatible destination
                invEndptRef (89),       -- invalid endpoint reference
                invTransNet (91),       -- invalid transit network selection
                manyAddptyReq (92),     -- too many add party requests
                aalParmUnsupp (93),     -- UNI 3.0:AAL paramteres cannot be
                                        -- supported
                infoElMssg (96),        -- mandatory info element is missing
                msgTypNotImpl (97),     -- message type not implemented
                infoElNotImpl (99),     -- info element not implemented
                invInfoEl (100),        -- invalid info element
                msgNotComp (101),       -- msg type not compatible with
                                        -- call st
                tmrRcvry (102),         -- recovery on timer expiry
                invMsgLen (104),        -- incorrect message length
                protErr (111),          -- protocol error, unspecified
                optElErr (127)          -- opt info el content error
                                        -- (non-std)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The last transmitted cause code for this port."
        ::= { atmSigLportEntry 18 }

atmSigLportQ93bLastDiagTx OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The last transmitted diagnostic code for this port."
        ::= { atmSigLportEntry 19 }

atmSigLportQ93bNumPduRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of Q93B PDU's received on this port."
        ::= { atmSigLportEntry 20 }
```

```
atmSigLportQ93bLastCauseRx OBJECT-TYPE
        SYNTAX  INTEGER {
                unallocNmb (1),         -- unallocated (unassigned) number
                noRoutTransnet (2),     -- no route to transit network
                noRoutDst (3),          -- no route to destination
                vccUnacpt (10),         -- UNI 3.0: VPCI/VCI unacceptable
                normCallClr (16),       -- UNI 3.1: normal call clearing
                usrBusy (17),           -- user busy
                nousrRsp (18),          -- no user response
                callRej (21),           -- call rejected
                nmbChng (22),           -- number changed
                callRejClir (23),       -- user rejects all calls with CLIR
                dstOutOrd (27),         -- destination out of order
                invNmbForm (28),        -- invalid number format
                rspStatEnq (30),        -- response to STATUS ENQUIRY
                normUnspec (31),        -- normal unspecified
                reqVccUnavail (35),     -- requested VPCI/VCI unavailable
                vccFail (36),           -- UNI 3.1: VPCI/VCI assignment
                                        -- failure

                rateUnavail1 (37),      -- UNI 3.1: user cell rate
                                        -- unavailable
                netOutOrd (38),         -- network out of order
                tmpFail (41),           -- Temporary failure
                accInfoDisc (43),       -- access info discarded
                noVccAvail (45),        -- no VPCI/VCI unavailable
                resUnavail (47),        -- resources unavailable,
                                        -- unspecified
                qosUnavail (49),        -- Quality of Service unavailable
                rateUnavail (51),       -- UNI 3.0: user cell rate
                                        -- unavailable
                bCapNotAuth (57),       -- bearer capability not authorized
                bCapUnavail (58),       -- bearer capability not available
                srvUnavail (63),        -- Service or option unavailable
                bCapNotImpl (65),       -- bearer capability not implemented
                combUnsupp (73),        -- unsupported comb. of traffic
                                        -- parameters
                aalParmUnsupp1 (78),    -- UNI 3.1: AAL paramteres cannot
                                        -- be supported
                invCallRef (81)         -- invalid call reference
                chanNotExst (82),       -- identified channel does not exist
                dstNotComp (88),        -- incompatible destination
                invEndptRef (89),       -- invalid endpoint reference
                invTransNet (91),       -- invalid transit network selection
                manyAddptyReq (92),     -- too many add party requests
                aalParmUnsupp (93),     -- UNI 3.0:AAL paramteres cannot be
                                        -- supported
                infoElMssg (96),        -- mandatory info element is missing
                msgTypNotImpl (97),     -- message type not implemented
                infoElNotImpl (99),     -- info element not implemented
                invInfoEl (100),        -- invalid info element
                msgNotComp (101),       -- msg type not compatible with
                                        -- call st
                tmrRcvry (102),         -- recovery on timer expiry
                invMsgLen (104),        -- incorrect message length
                protErr (111),          -- protocol error, unspecified
                optElErr (127)          -- opt info el content error
                                        -- (non-std)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The last transmitted cause code for this port."
        ::= { atmSigLportEntry 21 }

atmSigLportQ93bLastDiagRx OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The last transmitted diagnostic code for this port."
        ::= { atmSigLportEntry 22 }

atmSigLportQSaalMaxCC OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The maximum number of unacknowledged control PDU's before
                declaring a loss of connection."
        ::= { atmSigLportEntry 23 }

atmSigLportQSaalMaxPD OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The maximum number of PDU's transmitted before a POLL PDU
                is transmitted."
        ::= { atmSigLportEntry 24 }

atmSigLportQSaalMaxStat OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The maximum number of list elements in a STAT PDU."
        ::= { atmSigLportEntry 25 }

atmSigLportQSaalTPoll OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The protocol timer corresponding to the polling function,
                specified in milliseconds."
        ::= { atmSigLportEntry 26 }
```

```
atmSigLportQSaalTKeepalive OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The protocol timer corresponding to the keepalive
                function,
                specified in milliseconds."
        ::= { atmSigLportEntry 27 }

atmSigLportQSaalTNoResponse OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The protocol timer corresponding to the timeout function,
                specified in milliseconds."
        ::= { atmSigLportEntry 28 }

atmSigLportQSaalTCC OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The protocol timer corresponding to the transmission of
                control PDU's, specified in milliseconds."
        ::= { atmSigLportEntry 29 }

atmSigLportQSaalTIdle OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The protocol timer corresponding to the idle function for
                UNI 3.1 only, specified in milliseconds."
        ::= { atmSigLportEntry 30 }

atmSigLportQSaalNumDiscardTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of QSaal transmit discards fo this port."
        ::= { atmSigLportEntry 31 }

atmSigLportQSaalNumErrorTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of QSaal transmit errors on this port."
        ::= { atmSigLportEntry 32 }

atmSigLportQSaalNumPduTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of QSaal PDU's transmitted on this port."
        ::= { atmSigLportEntry 33 }

atmSigLportQSaalNumDiscardRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of QSaal receive discards on this port."
        ::= { atmSigLportEntry 34 }

atmSigLportQSaalNumErrorRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of QSaal receive errors on this port."
        ::= { atmSigLportEntry 35 }

atmSigLportQSaalNumPduRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of QSaal PDU's received on this port."
        ::= { atmSigLportEntry 36 }
```

The Circuit Group

```
--  The variables that configure and monitor circuits on a port.

cktTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF CktEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A table containing information about specific DLCIs,
                channels and corresponding circuits."
        ::= { ckt 1 }

cktEntry OBJECT-TYPE
        SYNTAX  CktEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "The information regarding a single Data Link Connection
                Identifier."
        INDEX { cktSrcIfIndex, cktSrcDlci }
        ::= { cktTable 1 }
```

```
CktEntry ::=
        SEQUENCE {
                cktSrcIfIndex
                    Index,
                cktSrcDlci
                    INTEGER,
                cktPriority
                    INTEGER,
                cktCir
                    INTEGER,
                cktBc
                    INTEGER,
                cktBe
                    INTEGER,
                cktDestNodeId
                    INTEGER,
                cktDestIfIndex
                    INTEGER,
                cktDestDlci
                    INTEGER,
                cktTos
                    INTEGER,
                cktOde
                    INTEGER,
                cktAdminStatus
                    INTEGER,
                cktCreationTime
                    TimeTicks,
                cktLastTimeChange
                    TimeTicks,
                cktVcState
                    INTEGER,
                cktDceState
                    INTEGER,
                cktDteStatus
                    INTEGER,
                cktRnr
                    INTEGER,
                cktNiDown
                    INTEGER,
                cktDteState
                    INTEGER,
                cktOperStatus
                    INTEGER,
                cktOutForward
                    INTEGER,
                cktRerouteCnt
                    INTEGER,
                cktVcPtr
                    OCTET STRING,
                cktHopCnt
                    INTEGER,
                cktPath
                    DisplayString,
                cktFailReason
                    INTEGER,
                cktFailNode
                    INTEGER,
                cktFailPort
                    INTEGER,
                cktMcastGroupId
                    INTEGER,
                cktMcastMemberList
                    OCTET STRING,
                cktMcastParentGroups
                    OCTET STRING,
                cktInFrames
                    Counter,
                cktInDEFrames
                    Counter,
                cktInODEFrames
                    Counter,
                cktInFECNFrames
                    Counter,
                cktInBECNFrames
                    Counter,
                cktInDiscards
                    Counter,
                cktInOctets
                    Counter,
                cktInDEOctets
                    Counter,
                cktInODEOctets
                    Counter,
                cktOutFrames
                    Counter,
                cktOutDEFrames
                    Counter,
                cktOutODEFrames
                    Counter,
                cktOutFECNFrames
                    Counter,
                cktOutBECNFrames
                    Counter,
                cktOutOctets
                    Counter,
                cktOutDEOctets
                    Counter,
                cktOutODEOctets
                    Counter,
                cktOutLostFrames
                    Counter,
                cktOutLostDEFrames
                    Counter,
                cktOutLostODEFrames
                    Counter,
                cktOutLostOctets
                    Counter,
```

```
cktOutLostDEOctets
    Counter,
cktOutLostODEOctets
    Counter,
cktRtMinDelay
    INTEGER,
cktRtMaxDelay
    INTEGER,
cktRtAvgDelay
    INTEGER,
cktDiagTestId
    INTEGER,
cktDiagTestRuns
    INTEGER,
cktHelloCounter
    INTEGER,
cktHelloAckCounter
    INTEGER,
cktDefinedPath
    OCTET STRING,
cktDefinedPathCount
    INTEGER,
cktDefinedPathEnable
    INTEGER,
cktDefinedPathAltOption
    INTEGER,
cktUsingDefinedPath
    INTEGER,
ckt0CIRCircuit
    INTEGER,
cktNotVirgin
    INTEGER,
cktInForward
    INTEGER,
cktBtusSeg
    INTEGER,
cktInSegmentsDiscards
    Counter,
cktAtmVPI
    INTEGER,
cktAtmVCI
    INTEGER,
cktType
    INTEGER,
cktSvcCallingParty
    OCTET STRING,
cktSvcCalledParty
    OCTET STRING,
cktSvcDuration
    TimeTicks,
cktSvcCause
    INTEGER,
cktXlatFlag
    INTEGER,
cktDestLaddr
    INTEGER,
cktSrcLaddr
    INTEGER,
cktLoop
    INTEGER,
cktRerouteBalance
    INTEGER,
cktCallingBackup
    INTEGER,
cktRCir
    INTEGER,
cktAtmQoS
    INTEGER,
cktAtmInCells
    Counter,
cktAtmOutCells
    Counter,
cktAtmInDiscardedClp0Cells
    Counter,
cktAtmInDiscardedClp1Cells
    Counter,
cktAtmVcType
    INTEGER,
cktAtmPCR
    INTEGER,
cktAtmSCR
    INTEGER,
cktAtmMBS
    INTEGER,
cktAtmInPassedClp0Cells
    Counter,
cktAtmInPassedClp1Cells
    Counter,
cktAtmInTaggedCells
    Counter,
cktAtmOutClp0Cells
    Counter,
cktAtmOutClp1Cells
    Counter,
cktAtmRQoS
    INTEGER,
cktAtmTfdType
    INTEGER,
cktAtmRTfdType
    INTEGER,
cktAtmTfdParam1
    INTEGER,
cktAtmTfdParam2
    INTEGER,
cktAtmTfdParam3
    INTEGER,
cktAtmRTfdParam1
    INTEGER,
```

```
cktAtmRTfdParam2
    INTEGER,
cktAtmRTfdParam3
    INTEGER,
cktAtmFrameIWF
    INTEGER,
cktAtmUserPlane
    INTEGER,
cktRBc
    INTEGER,
cktRBe
    INTEGER,
cktOamLoopbackDirection
    INTEGER,
cktOamLoopbackType
    INTEGER,
cktOamLoopbackHops
    INTEGER,
cktOamLoopbackCount
    INTEGER,
cktOamLoopbackReceived
    Counter,
cktOamLoopbackTimeouts
    Counter,
cktOamLoopbackReceivedHigh
    INTEGER,
cktOamLoopbackReceivedLow
    INTEGER,
cktOamLoopbackReceivedAvg
    INTEGER,
cktOamAlarmDisable
    INTEGER,
cktShaperId
    INTEGER,
cktOspfCtd
    INTEGER,
cktOspfCdv
    INTEGER,
cktOutPort
    INTEGER,
cktOutVc
    INTEGER,
cktRVc
    INTEGER,
cktEntryType
    INTEGER,
cktDiagStr
    OCTET STRING,
cktSvcAalType
    INTEGER,
cktSvcBBearerClass
    INTEGER,
cktSvcBBearerClippingSusc
    INTEGER,
cktSvcBBearerTmgReq
    INTEGER,
cktSvcBBearerTfcType
    INTEGER,
cktAtmUPCEnable
    INTEGER,
cktRPriority
    INTEGER,
cktRtPriority
    INTEGER,
cktDeltaBc
    INTEGER,
cktDeltaBe
    INTEGER,
cktDeltaRBc
    INTEGER,
cktDeltaRBe
    INTEGER,
cktRedFrPcnt
    INTEGER,
cktRedFrRPcnt
    INTEGER,
cktRateEnforceSchm
    INTEGER,
cktRateEnforceRSchm
    INTEGER,
cktROde
    INTEGER,
cktPrivateNet
    INTEGER,
cktPrivateNetOverflow
    INTEGER,
cktCustomerID
    INTEGER,
cktAtmCDVT
    INTEGER,
cktNdcEnable
    INTEGER,
cktInterworkingFrToAtmCLP
    INTEGER,
cktInterworkingFrToAtmDe
    INTEGER,
cktNrtsCLP1
    INTEGER,
cktNrtsDiscardClp0
    Counter,
cktNrtsDiscardClp1
    Counter,
cktMPEnableAMF
    INTEGER,
cktMPEligible
    INTEGER,
cktMPForcedCaller
    INTEGER,
```

```
                cktMPForcedCallee
                    INTEGER,
                cktFrameSize
                    INTEGER,
                cktRFrameSize
                    INTEGER,
                cktRNrtsCLP1
                    INTEGER
                }

cktSrcIfIndex OBJECT-TYPE
        SYNTAX  Index
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The ifIndex value of the corresponding ifEntry. The
                ifEntry  must be of type supporting this circuit. The
                lportProtocol must be either fr or some other valid UNI/
                NNI."
        ::= { cktEntry 1 }

cktSrcDlci OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The DLCI used as the key for the circuit. For local DLCI
                significance, this is the local DLCI. For Global DLCI
                significance, this is the remote DLCI.  For ATM circuits,
                the VPI (most significant 16 bits) and VCI (least
                significant 16 bits) are concatenated to form this value.
                Significant bounds for setting this index are based on the
                LMI rev type on this ifEntry:
                    Lower  Upper  Type
                    0      4096   Disabled
                    16     1007   LMIREV1
                    16     991    All other FR types."
        ::= { cktEntry 2 }

cktPriority OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The priority level (0 through 3) for this circuit.
                When port service type is defined as 'mono-class', this
                 priority means the forward priority of the circuit.
                When port service type is defined as 'multi-class', this
                 priority means the discard priority of the circuit. "
 ::= { cktEntry 3 }

cktCir OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The average number of user data (bits) that the network
                agrees to transfer over the circuit in one direction,
                measured over the measurement interval T = cktBc/cktCir."
        ::= { cktEntry 4 }

cktBc OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The maximum amount of data (bits) that the network agrees
                to transfer over the circuit under normal conditions,
                during the measurement interval."
        ::= { cktEntry 5 }

cktBe OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The maximum amount of uncommitted data (bits) that the
                network will attempt to transfer over the circuit during
                the measurement interval.  By default, if not configured
                when creating the entry, the Excess Information Burst Size
                is set to the value of ifSpeed."
        ::= { cktEntry 6 }

cktDestNodeId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The destination node ID for this circuit."
        ::= { cktEntry 7 }

cktDestIfIndex OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The interface identifier at the destination node for this
                circuit."
        ::= { cktEntry 8 }
```

```
cktDestDlci OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The DLCI which is the destination of the key DLCI. For
                local DLCI significance, this is the remote DLCI since the
                key DLCI is the local DLCI. For global significance, this
                is the local DLCI since the key DLCI is the remote DLCI.
                For ATM circuits, the VPI (most significant 16 bits)
                and VCI (least significant 16 bits) are concatenated to
                form this value."
        ::= { cktEntry 9 }

cktTos OBJECT-TYPE
        SYNTAX  INTEGER {
                committed (1),
                shared (2)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The type of service for the circuit. Only committed is
                supported in the FCS."
        ::= { cktEntry 10 }
cktOde OBJECT-TYPE
        SYNTAX  INTEGER {
                    off (0),
                    on (1)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "This variable states whether graceful discard is enabled
                for the ckt."
        ::= { cktEntry 11 }

cktAdminStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                    invalid (0),
                    down (1),
                    up (2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The desired state of the ckt."
        DEFVAL { up }
        ::= { cktEntry 12 }

cktCreationTime OBJECT-TYPE
        SYNTAX  TimeTicks
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The value of sysUpTime when the circuit was created
                (activated)."
        ::= { cktEntry 13 }

cktLastTimeChange OBJECT-TYPE
        SYNTAX  TimeTicks
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The sysUpTime value when the circuit was last changed to
                the current state."
        ::= { cktEntry 14 }

cktVcState OBJECT-TYPE
        SYNTAX  INTEGER {
                    invalid (0),
                    inactive (1),
                    retry (2),
                    calling (3),
                    wcbdeact(4),
                    wcbdelete(5),
                    active (6),
                    svcall (7),
                    svclr (8),
                    backedup (9),
                    wcbbkdp (10),
                    wcbreact (11),
                    slowretry (12)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The current state of the PVC segment in the Cascade
                network."
        ::= { cktEntry 15 }

cktDceState OBJECT-TYPE
        SYNTAX  INTEGER {
                invalid (0),
                inactive (1),
                active (2)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The DCE state of the circuit."
        ::= { cktEntry 16 }
```

```
cktDteStatus OBJECT-TYPE
        SYNTAX      INTEGER {
                invalid (0),
                inactive (1),
                active (2)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The DTE status of the circuit."
        ::= { cktEntry 17 }

cktRnr OBJECT-TYPE
        SYNTAX  INTEGER {
                recvnotready (1),
                recvready (0)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The receiver's readiness for accepting data flow."
        ::= { cktEntry 18 }

cktNiDown OBJECT-TYPE
        SYNTAX  INTEGER {
                niup (0),
                nidown (1)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The state of the network interface."
        ::= { cktEntry 19 }

cktDteState OBJECT-TYPE
        SYNTAX  INTEGER {
                invalid (0),
                inactive (1),
                active (2)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The DTE state of the circuit."
        ::= { cktEntry 20 }

cktOperStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                invalid (0),
                inactive (1),
                active (2)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The current operational status of the entire PVC."
        ::= { cktEntry 21 }

cktOutForward OBJECT-TYPE
        SYNTAX  INTEGER {
                off (0),
                on (1)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "If the value is 1, it means the outbound flow is on."
        ::= { cktEntry 22 }

cktRerouteCnt OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The reroute count."
        ::= { cktEntry 23 }

cktVcPtr OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "An 8-byte Octect string indicating the vc pointer."
        ::= { cktEntry 24 }

cktHopCnt OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The count of hops along the circuit path. (Max is 16)"
        ::= { cktEntry 25 }
```

```
cktPath OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The circuit path consisting a sequence of outbound
                interface indexes at nodes along the established circuit.
                The format is interface:interface:interface......"
        ::= { cktEntry 26 }

cktFailReason OBJECT-TYPE
        SYNTAX  INTEGER {
                    none (0),
                    admindown (1),
                    novcbuff (2),
                    nobw (3),
                    noroute (4),
                    timeout (5),
                    nopdubuff (6),
                    nodest (7),
                    trkrnr (8),
                    trkdown (9),
                    balancereroute (10),
                    dead(11),
                    defpathreroute(12),
                    nidown(13),
                    otherpvcsegdown(14),
                    otherpvcsegrnr(15),
                    usingaltpathwarning(16),
                    iopdown(17),
                    numsgbuffer(18),
                    noport(19),
                    misconfig(20),
                    svcsetupfail(21),
                    srcbackedup(22),
                    srcunknown(23),
                    dstunknown(24),
                    bkpdlcicollision(25),
                    oldrevinpath(26),
                    smdsmgmttrunk(27),
                    nevercalled(28),
                    bothendptbackup(29),
                    pvcroutemgttrunk(30),
                    nomultipointparent(31),
                    pvcroutefail(32),
                    novpivci(33),
                    svcuserclear(34),
                    pathregfailed(35),
                    noatmchan(36),
                    norevbw(37),
                    internalreset(38)
                   highprivcinpath(39),
                    nopribw(40)
                    }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The reason for the PVC establishment failure."
        ::= { cktEntry 27 }

cktFailNode OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The node which causes the PVC failure."
        ::= { cktEntry 28 }

cktFailPort OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The port on the fail node which causes the PVC failure."
        ::= { cktEntry 29 }

cktMcastGroupId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Multicast group identifier (1..32).
                This field is 0 for unicast PVCs."
        ::= { cktEntry 30 }

cktMcastMemberList OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-write
        STATUS  mandatory

        DESCRIPTION
                "A list of operations and the affected unicast DLCIs
                (lportId, cktId)s in a multicast group.  The syntax is as
                follows:
                [op:dlci,dlci,...,op:dlci,dlci,...] where op can either be
                delete (0) or add (1)"
        ::= { cktEntry 31 }

cktMcastParentGroups OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "An 8-byte Octect string indicating the parent groups this
                DLCI belongs to."
        ::= { cktEntry 32 }
```

```
cktInFrames OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of inbound frames since the last reset."
        ::= { cktEntry 33 }

cktInDEFrames OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of inbound DE-marked frames since the last
                reset."
        ::= { cktEntry 34 }

cktInODEFrames OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of inbound ODE-marked frames since the last
                reset."
        ::= { cktEntry 35 }

cktInFECNFrames OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of inbound frames indicating forward congestion
                since the last reset."
        ::= { cktEntry 36 }

cktInBECNFrames OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of inbound frames indicating backward
                congestion since the last reset."
        ::= { cktEntry 37 }

cktInDiscards OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Number of inbound frames discarded by rate enforcement."
        ::= { cktEntry 38 }

cktInOctets OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of inbound octets since the last reset."
        ::= { cktEntry 39 }

cktInDEOctets OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of inbound DE-marked octets since the last
                reset."
        ::= { cktEntry 40 }

cktInODEOctets OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of inbound ODE-marked octets since the last
                reset."
        ::= { cktEntry 41 }

cktOutFrames OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of outbound frames since the last reset."
        ::= { cktEntry 42 }

cktOutDEFrames OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of outbound DE-marked frames since the last
                reset."
        ::= { cktEntry 43 }

cktOutODEFrames OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of outbound ODE-marked frames since the last
                reset."
        ::= { cktEntry 44 }
```

```
cktOutFECNFrames OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS      mandatory
        DESCRIPTION
                "The number of outbound FECN-marked frames since the last
                reset."
        ::= { cktEntry 45 }

cktOutBECNFrames OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of outbound BECN-marked frames since the last
                reset."
        ::= { cktEntry 46 }

cktOutOctets OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of outbound octets since the last reset."
        ::= { cktEntry 47 }

cktOutDEOctets OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of outbound DE-marked octets since the last
                reset."
        ::= { cktEntry 48 }

cktOutODEOctets OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of outbound ODE-marked octets since the last
                reset."
        ::= { cktEntry 49 }

cktOutLostFrames OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of outbound frames that have been lost since
                the last reset."
        ::= { cktEntry 50 }

cktOutLostDEFrames OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of outbound DE-marked frames that have been
                lost since the last reset."
        ::= { cktEntry 51 }

cktOutLostODEFrames OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of outbound ODE-marked frames that have been
                lost since the last reset."
        ::= { cktEntry 52 }

cktOutLostOctets OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of outbound octets that have been lost since
                the last reset."
        ::= { cktEntry 53 }

cktOutLostDEOctets OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of outbound DE-marked octets that have been
                lost since the last reset."
        ::= { cktEntry 54 }

cktOutLostODEOctets OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of outbound ODE-marked octets that have been
                lost since the last reset."
        ::= { cktEntry 55 }

cktRtMinDelay OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The minimum round-trip delay (micro-seconds)."
        ::= { cktEntry 56 }
```

```
cktRtMaxDelay OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The maximum round-trip delay (micro-seconds)."
        ::= { cktEntry 57 }

cktRtAvgDelay OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The average round-trip delay (micro-seconds)."
        ::= { cktEntry 58 }

cktDiagTestId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Identification for the diagnostics tests to be run."
        ::= { cktEntry 59 }

cktDiagTestRuns OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of passes of the diagnostics tests to be run.
                The default value is 1."
        ::= { cktEntry 60 }

cktHelloCounter OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Number of PVC hello pdu frames received in the VC entry
                of the called side."
        ::= { cktEntry 61 }

cktHelloAckCounter OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Number of PVC hello Ack pdu frames received in the VC
                entry of the calling side."
        ::= { cktEntry 62 }

cktDefinedPath OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "User defined path. in the format of Nx,x,x,Nx,.... If x is
                not prefixed with 'N', x is an interface ID. If x is
                prefixed with ‘N‘, x is a node ID."
        ::= { cktEntry 63 }

cktDefinedPathCount OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "User defined path hop count."
        ::= { cktEntry 64 }

cktDefinedPathEnable OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "If set to 1, it means there is an user defined path for
                this circuit and it is enabled."
        ::= { cktEntry 65 }

cktDefinedPathAltOption OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "If set to 1, it means that if the user defined path fails,
                use the ospf-determined path."
        ::= { cktEntry 66 }

cktUsingDefinedPath OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "If set to 1, it indicates the PVC is currently using the
                defined path."
        ::= { cktEntry 67 }

cktTryAltPath OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "If set to 1, it means the inactive PVC will be activated
                on the ospf-determined path in the next retry."
        ::= { cktEntry 68 }
cktNotVirgin OBJECT-TYPE
```

```
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "If set to 1, it means this VC entry has been successfuly
                activated before."
        ::= { cktEntry 69 }

cktInForward OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "If the value is 1, it means the inbound flow is OK."
        ::= { cktEntry 70 }

cktBtusSeg OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of Bus xmit units per data segment"
        ::= { cktEntry 71 }

cktInSegmentsDiscards OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Number of segments received that were discarded because
                of frame reassembly errors."
        ::= { cktEntry 72 }

cktAtmVPI OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "VPI value in the ATM cell header:
                 ATM DXI with HSSI IOP VPI (4 lsb bit) range: 0 - 15
                 ATM UNI DS3/E3 IOP    VPI (4 lsb bit) range: 0 - 15
                 T1 ATM 4 lsb  range 0-15
                 ATM-IWU STM-1/STS-3c  VPI (<=12 lsb bit) range: 0-4095"
        ::= { cktEntry 73 }

cktAtmVCI OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "VCI value in the ATM cell header:
                 ATM DXI with HSSI IOP VCI (6 lsb bit) range: 0 - 63
                 ATM UNI DS3/E3 IOP    VCI (8 lsb bit) range: 0 - 255
                T1 ATM 4 lsb          VCI (6 lsb bit) range  0 - 255
                ATM-IWU STM-1/STS-3c  VCI (<=12 lsb bit) range: 0-4095"
        ::= { cktEntry 74 }

cktType OBJECT-TYPE
        SYNTAX  INTEGER  {
                pvc (1),
                svc (2)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "1 if a permanent virtual circuit; 2 if a switched virtual
                circuit."
        ::= { cktEntry 75 }

cktSvcCallingParty OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The calling party number if this is a switched virtual
                circuit."
                ::= { cktEntry 76 }

cktSvcCalledParty OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The called party number if this is a switched virtual
                circuit."
        ::= { cktEntry 77 }

cktSvcDuration OBJECT-TYPE
        SYNTAX  TimeTicks
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The duration since the SVC circuit has been running."
        ::= { cktEntry 78 }
```

```
cktSvcCause OBJECT-TYPE
        SYNTAX  INTEGER {
                unalloc-nmb     (1),        -- unallocated (unassigned)
                                             number
                no-route-transnet(2),       -- no route to transit network
                no-route-dest   (3),        -- no route to destination
                vcc-unacceptable-30(10),    -- UNI 3.0: VPCI/VCI unacceptable
                normal-call-clr-31(16),     -- UNI 3.1: normal call clearing
                user-busy       (17),       -- user busy
                no-user-response(18),       -- no user response
                call-reject     (21),       -- call rejected
                nmb-changed     (22),       -- number changed
                call-reject-clir(23),       -- user rejects all calls with
                                               CLIR
                dest-out-of-order(27),      -- destination out of order
                invalid-nmb-format(28),     -- invalid number format
                response-stat-enq(30),      -- response to STATUS ENQUIRY
                normal-unspecified(31),     -- normal unspecified
                req-vcc-unavailable(35),    -- requested VPCI/VCI unavailable
                vcc-fail-31     (36),       -- UNI 3.1: VPCI/VCI assignment
                                               failure
                rate-unavail-31 (37),       -- UNI 3.1: user cell rate
                                               unavailable
                network-out-of-order(38),   -- network out of order
                temp-fail       (41),       -- Temporary failure
                access-info-discard(43),    -- access info discarded
                no-vcc-available(45),       -- no VPCI/VCI unavailable
                resources-unavailable(47),-- resources unavailable,
                                               unspecified
                qos-unavailable (49),       -- Quality of Service unavailable
                rate-unavailable-30(51),    -- UNI 3.0: user cell rate
                                               unavailable
                b-cap-not-authorized(57),   -- bearer capability not
                                               authorized
                b-cap-unavailable(58),      -- bearer capability not
                                               available
                service-unavailable(63),    -- Service or option unavailable
                b-cap-not-implemented(65),-- bearer capability not
                                               implemented
                combination-unsupported(73),-- unsupported comb. of
                                                 traffic parameters
                aal-params-unsupp-31(78),   -- UNI 3.1: AAL paramteres cannot
                                               be supported
                invalid-call-reference(81),--invalid call reference
                no-channel      (82),       -- identified channel does not
                                               exist
                dest-incompatible(88),      -- incompatible destination
                invalid-endpoint-ref(89),   -- invalid endpoint reference
                invalid-transit-net(91),    -- invalid transit network
                                               selection
                too-many-add-pty-req(92),   -- too many add party requests
                aal-params-unsupp-30(93),   -- UNI 3.0:AAL paramteres cannot
                                               be supported
                info-element-missing(96),   -- mandatory info element is
                                                missing
                msg-type-not-imp(97),       -- message type not implemented
                info-element-not-imp(99),   -- info element not implemented
                invalid-info-element(100),-- invalid info element
                message-not-compatible(101),-- msg type not compatible
                                                with call st
                timer-recovery (102),       -- recovery on timer expiry
                invalid-message-len(104),   -- incorrect message length
                protocol-error (111),       -- protocol error, unspecified
                optional-element-error(127)-- opt info el content error
                                                (non-std)
        }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "SVC cause number."
        ::= { cktEntry 79 }

cktXlatFlag OBJECT-TYPE
        SYNTAX  INTEGER  {
                no-translation (0),
                rfc1483 (1)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "0 if RFC1490 or no translation; 1 if RFC1483 translation."
        ::= { cktEntry 80 }

cktDestLaddr OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The destination logical address of the circuit."
        ::= { cktEntry 81 }

cktSrcLaddr OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The source logical address of the circuit."
        ::= { cktEntry 82 }
```

```
cktLoop OBJECT-TYPE
        SYNTAX  INTEGER {
                disabled(0),
                local(1),
                remote(2),
                both(3)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Setting this variable controls the loopback status of the
                given circuit endpoint."
        ::= { cktEntry 83 }

cktRerouteBalance OBJECT-TYPE
        SYNTAX INTEGER {
                enabled(0),
                disabled(1)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Setting this variable controls the use of rerouting to
                balance link usage."
        ::= { cktEntry 84 }

cktCallingBackup OBJECT-TYPE
        SYNTAX  INTEGER {
                false(0),
                true(1)
                }
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Indicates if a caller endpoint is calling a backup PVC."
        ::= { cktEntry 85 }

cktRCir OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The average number of user data (bits) that the network
                agrees to transfer over the circuit in the opposite
                direction, measured over the measurement interval T =
                cktBc/cktCir."
        ::= { cktEntry 86 }

cktAtmQoS OBJECT-TYPE
        SYNTAX INTEGER {
                cbr  (1),
                vbr1 (2),
                vbr2 (3),
                ubr (4),
                unspecified (5)
        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The Quality of Service of the ATM Circuit. Valid only for
                 ATM type circuits."
        ::= { cktEntry 87 }

cktAtmInCells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The Number of ATM cells received on a VC (VPC or VCC)."
        ::= { cktEntry 88 }

cktAtmOutCells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The Number of ATM cells transmitted on a VC (VPC or
                VCC).."
        ::= { cktEntry 89 }

cktAtmInDiscardedClp0Cells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The Number of ATM CLP0 cells received and discarded on a
                VC (VPC or VCC).."
        ::= { cktEntry 90 }

cktAtmInDiscardedClp1Cells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The Number of ATM CLP1 cells received and discarded on a
                VC (VPC or VCC)."
        ::= { cktEntry 91 }
```

```
cktAtmVcType OBJECT-TYPE
        SYNTAX  INTEGER {
                vpc  (1),
                vcc  (2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Type of ATM connection (VPC or VCC)."
        ::= { cktEntry 92 }

cktAtmPCR OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The Peak Cell Rate  measured in cells/second at which
                cells are transmitted for this circuit."
        ::= { cktEntry 93 }

cktAtmSCR OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The Sustainable Cell Rate is the average transmission rate
                in cells per second for this circuit. It must be less than
                or equal to the Peak Cell Rate."
        ::= { cktEntry 94 }

cktAtmMBS OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The Maximum Burst size determines the maximum number of
                cells that can be transmitted at the peak cell rate."
        ::= { cktEntry 95 }

cktAtmInPassedClp0Cells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The Number of ATM CLP 0 cells received and passed UPC on
                a VC (VPC or VCC)."
        ::= { cktEntry 96 }

cktAtmInPassedClp1Cells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The Number of ATM CLP 1 cells received and passed UPC on
                a VC (VPC or VCC)."
        ::= { cktEntry 97 }

cktAtmInTaggedCells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The Number of ATM cells received and tagged on a VC (VPC
                or VCC)."
        ::= { cktEntry 98 }

cktAtmOutClp0Cells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The Number of ATM CLP 0 cells transmitted on a VC (VPC or
                VCC).."
        ::= { cktEntry 99 }

cktAtmOutClp1Cells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The Number of ATM CLP 1 cells transmitted on a VC (VPC or
                VCC).."
        ::= { cktEntry 100 }

cktAtmRQoS OBJECT-TYPE
        SYNTAX  INTEGER {
                cbr  (1),
                vbr1 (2),
                vbr2 (3),
                vbr3 (4)
        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The Quality of Service of the ATM connection in the
                opposite direction."
        ::= { cktEntry 101 }
```

```
cktAtmTfdType OBJECT-TYPE
        SYNTAX  INTEGER {
                pcr-0-01 (1),
                pcr-0-01-tag (2),
                pcr-01-scr-0-mbs-0 (3),
                pcr-01-scr-0-mbs-0-tag (4),
                pcr-01 (5),
                pcr-01-scr-01-mbs-01(6),
                pcr-01-bestEffort(7)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The combination of traffic parameters."
        ::= { cktEntry 102 }

cktAtmRTfdType OBJECT-TYPE
        SYNTAX  INTEGER {
                pcr-0-01 (1),
                pcr-0-01-tag (2),
                pcr-01-scr-0-mbs-0 (3),
                pcr-01-scr-0-mbs-0-tag (4),
                pcr-01 (5),
                pcr-01-scr-01-mbs-01 (6),
                pcr-01-bestEffort(7)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The combination of traffic parameters in opposite
                direction."
        ::= { cktEntry 103 }

cktAtmTfdParam1 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The traffic descriptor parameter 1."
        ::= { cktEntry 104 }
cktAtmTfdParam2 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The traffic descriptor parameter 2."
        ::= { cktEntry 105 }

cktAtmTfdParam3 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The traffic descriptor parameter 3."
        ::= { cktEntry 106 }

cktAtmRTfdParam1 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The traffic descriptor parameter 1 for opposite
                direction."
        ::= { cktEntry 107 }

cktAtmRTfdParam2 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The traffic descriptor parameter 3 for opposite
                direction."
        ::= { cktEntry 108 }

cktAtmRTfdParam3 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The traffic descriptor parameter 3 for opposite
                direction."
        ::= { cktEntry 109 }

cktAtmFrameIWF OBJECT-TYPE
        SYNTAX  INTEGER {
                    default(1),
                    iwf(2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Frame Relay/ATM Interworking Function indicator."
        ::= { cktEntry 110 }
```

```
cktAtmUserPlane OBJECT-TYPE
        SYNTAX  INTEGER {
                    point-to-point(1),
                    point-to-multipoint(2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Type of connectivity for this circuit."
        ::= { cktEntry 111 }

cktRBc OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The maximum amount of data (bits) that the network agrees
                to transfer over the circuit in the opposite direction
                under normal conditions, during the measurement interval."
        ::= { cktEntry 112 }

cktRBe OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The maximum amount of uncommitted data (bits) that the
                network will attempt to transfer over the circuit in the
                opposite direction during the measurement interval.  By
                default, if not configured when creating the entry, the
                Excess Information Burst Size is set to the value of
                ifSpeed."
        ::= { cktEntry 113 }

cktOamLoopbackDirection OBJECT-TYPE
        SYNTAX  INTEGER {
                local(1),
                remote(2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Which direction to send the loopback cell.  If 'local',
                send out this circuit entry's port.  If 'remote', send out
                the remote circuit entry's port."
        ::= { cktEntry 114}

cktOamLoopbackType OBJECT-TYPE
        SYNTAX  INTEGER {
                segment(1),
                end-to-end(2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Type of oam loopback to send - either segment or end to
                end."
        ::= { cktEntry 115}

cktOamLoopbackHops OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of hops the loopback cell should traverse inside
                the Cascade network before being echoed back by the far
                Cascade device."
        ::= { cktEntry 116}

cktOamLoopbackCount OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Number of oam loopback cells left to send this session.
                Setting this from zero to non-zero starts the loopback
                session."
        ::= { cktEntry 117}

cktOamLoopbackReceived OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Number of oam responses received for this loopback
                session."
        ::= { cktEntry 118}

cktOamLoopbackTimeouts OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Number of times a loopback response was not received
                within the timeout period during this loopback session."
        ::= { cktEntry 119}
```

```
cktOamLoopbackReceivedHigh OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Highest response time (in milliseconds) of an oam loopback
                response this loopback session."
        ::= { cktEntry 120}

cktOamLoopbackReceivedLow OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Lowest response time (in milliseconds) of an oam loopback
                response this loopback session."
        ::= { cktEntry 121}

cktOamLoopbackReceivedAvg OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Average response time (in milliseconds) of an oam loopback
                response this loopback session."
        ::= { cktEntry 122}

cktOamAlarmDisable OBJECT-TYPE
        SYNTAX  INTEGER {
                enabled(1),
                disabled(2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "If disabled, then don't generate oam alarms if this
                circuit is down."
        ::= { cktEntry 123}

cktShaperId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The shaper to be used for this virtual connection
                interworking with frame relay:

                1-port ATM-IWU STM-1/STS-3c card
                -------------------------------
                Values: 1..16"
        ::= { cktEntry 124 }

cktOspfCtd OBJECT-TYPE
        SYNTAX  INTEGER (1..100)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
               "The allowed cell delay:

                4-port CBR E1/DS1 card
                ----------------------
                The maximum allowed cell delay in msec"
        ::= { cktEntry 125 }

cktOspfCdv OBJECT-TYPE
        SYNTAX  INTEGER (100..1000)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The allowed cell delay variation:

                4-port CBR E1/DS1 card
                ----------------------
                The maximum allowed cell delay variation in msec"
        ::= { cktEntry 126 }

cktOutPort OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The outgoing port number for the adjacent VC entry in
                this switch."
        ::= { cktEntry 127 }

cktOutVc OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The adjacent VC entry corresponding to this circuit
                accross the bus."
        ::= { cktEntry 128 }

cktRVc OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The adjacent VC entry corresponding to this circuit
                accross the trunk."
        ::= { cktEntry 129 }
```

```
cktEntryType OBJECT-TYPE
        SYNTAX  INTEGER {
                    fr-user(1),
                    as-trunk(2),
                    on-trunk(3),
                    lmi(4),
                    multicast(5),
                    mgmt(6),
                    smds(7),
                    split-multicast(8),
                    control(9),
                    atm-user(10),
                    atm-leaf(11)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The internal type of circuit entry allocated by the
                circuit manager."
        ::= { cktEntry 130 }

cktDiagStr OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Internal diagnostic information."
        ::= { cktEntry 131 }

cktSvcAalType OBJECT-TYPE
        SYNTAX  INTEGER {
                aal1 (1),
                unspecified (2),
                aal3-4 (3),
                user-defined (4),
                aal5 (5)
        }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The AAL type for this SVC."
        ::= { cktEntry 132 }

cktSvcBBearerClass OBJECT-TYPE
        SYNTAX  INTEGER {
                unspecified (1),
                class-A (2),
                class-C (3),
                class-X (4)
        }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The Broadband bearer capability class for this SVC."
        ::= { cktEntry 133 }

cktSvcBBearerClippingSusc OBJECT-TYPE
        SYNTAX  INTEGER {
                unspecified (1),
                not-susceptible (2),
                susceptible (3)
        }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The Broadband bearer capability clipping susceptiblity
                for this SVC."
        ::= { cktEntry 134 }

cktSvcBBearerTmgReq OBJECT-TYPE
        SYNTAX  INTEGER {
                not-indicated (1),
                end-to-end (2),
                not-end-to-end (3)
        }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The Broadband bearer capability timing requirements for
                this SVC."
        ::= { cktEntry 135 }

cktSvcBBearerTfcType OBJECT-TYPE
        SYNTAX  INTEGER {
                not-indicated (1),
                cbr (2),
                vbr (3)
        }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The Broadband bearer capability traffic type for this
                SVC."
        ::= { cktEntry 136 }
```

```
cktAtmUPCEnable OBJECT-TYPE
        SYNTAX  INTEGER {
                disabled(1),
                enabled (2)
        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
        "Enable ATM UPC Function."
        ::= { cktEntry 137 }


cktRPriority OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The priority level (0 through 3) for this circuit in the
                opposite direction.When port service type is defined as
                'mono-class', this priority means the forward priority of
                the circuit.When port service type is defined as 'multi-
                class', this priority means the discard priority of the
                circuit."
        ::= { cktEntry 138 }

cktRtPriority OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The routing priority level of the circuit."
        ::= { cktEntry 139 }

cktDeltaBc OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The maximum number of bits that the network agree to
                transfer over the circuit as committed bits during the
                measurement interval under the CONDITION that the circuit
                still has POSITIVE committed bit (Bc) credits before
                receiving a frame but will have NEGATIVE Bc credits after
                accepting the frame. The range of this variable is 0 to
                65,528 bits. By default, if not configured when
                creating the entry, it is set to 65,528 bits."
        ::= { cktEntry 140 }

cktDeltaBe OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The maximum number of bits that the network agree to
                transfer over the circuit as allowed excess bits during the
                measurement interval under the CONDITION that the circuit
                still has POSITIVE excess bit (Be) credits before
                receiving a frame but will have NEGATIVE Be credits after
                accepting the frame. The range of this variable is 0 to
                65,528 bits. By default, if not configured when creating
                the entry, it is set to 65,528 bits."
        ::= { cktEntry 141 }

cktDeltaRBc OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The maximum number of bits that the network agree to
                transfer over the circuit as committed bits in the
                opposite direction during the measurement interval under
                the CONDITION that the circuit still has POSITIVE
                committed bit (Bc) credits before receiving a frame but
                will have NEGATIVE Bc credits after accepting the frame.
                The range of this variable is 0 to 8 Kbits. By default, if
                not configured when creating the entry, it is set to 0."
        ::= { cktEntry 142 }

cktDeltaRBc OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The maximum number of bits that the network agree to
                transfer over the circuit as committed bits in the
                opposite direction during the measurement interval under
                the CONDITION that the circuit still has POSITIVE
                committed bit (Bc) credits before receiving a frame but
                will have NEGATIVE Bc credits after accepting the frame.
                The range of this variable is 0 to 65,528 bits. By
                default, if not configured when creating the entry, it is
                set to 65,528 bits."
        ::= { cktEntry 142 }
```

```
cktDeltaRBe OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The maximum number of bits that the network agree to
                transfer over the circuit as allowed excess bits in the
                opposite direction during the measurement
                interval under the CONDITION that the circuit still has
                POSITIVE excess bit (Be) credits before receiving a frame
                but will have NEGATIVE Be credits after accepting the
                frame. The range of this variable is 0 to 65,528 bits. By
                default, if not configured when creating the entry, it is
                set to 65,528 bits."
        ::= { cktEntry 143 }

cktRedFrPcnt OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The percentage of RED frame bits that are allowed to
                enter the network, during the measurement interval, under
                non-congestion condition. Its value range is 1 - 100 and
                default value is 0. Its calculation is as follows:
                                        allowed RED frame bits
                    cktRedFrPcn = ------------------------------
                                  Bc+Be+allowed RED frame bits"
        ::= { cktEntry 144 }

cktRedFrRPcnt OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The percentage of RED frame bits that are allowed to
                enter the network in the opposite direction,
                during the measurement interval, under non-congestion
                condition. Its value range is 0 - 100 and default value is
                100.
                Its calculation is as follows:

                cktRedFrRPcn = (allowed RED frame bits)/(rBc+rBe+allowed
                               RED frame bits)"
        ::= { cktEntry 145 }

cktRateEnforceSchm OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Variable used to indicate the rate enforcement scheme
                employed.
                0: Slide window rate enforcement scheme
                1: Jump window rate enforcement scheme
                2: Simple rate enforcement scheme
                    Default value is 2."
        ::= { cktEntry 146 }

cktRateEnforceRSchm OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Variable used to indicate the rate enforcement scheme
                employed in the opposite direction.
                0: Slide window rate enforcement scheme
                1: Jump window rate enforcement scheme
                2: Simple rate enforcement scheme
                Default value is 2."
        ::= { cktEntry 147 }

cktORde OBJECT-TYPE
        SYNTAX  INTEGER {
                    off (0),
                    on (1)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "This variable states whether graceful discard is enabled
                for the ckt in the opposite direction."
        ::= { cktEntry 148 }

cktPrivateNet OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "If non-zero, indicates the private network that the
                circuit belongs to. If 0, the circuit has access to the
                entire public portion of the network."
        ::= { cktEntry 149 }
```

```
cktPrivateNetOverflow OBJECT-TYPE
        SYNTAX  INTEGER {
                    restrict (0),
                    use-public (1)
                 }
       ACCESS  read-write
       STATUS  mandatory
       DESCRIPTION
               "Indicates how private network circuits are handled,
               when the resources of the network have become exahusted.
               If set to use-public, the resources of the public network
               can be used during overflow conditions."
        ::= { cktEntry 150 }

cktCustomerID OBJECT-TYPE
       SYNTAX INTEGER
       ACCESS read-write
       STATUS mandatory
       DESCRIPTION
               "The Customer that owns this circuit. For Virtual
               Private Networking Support."
       ::= { cktEntry 151 }

cktAtmCDVT OBJECT-TYPE
       SYNTAX INTEGER
       ACCESS read-write
       STATUS mandatory
       DESCRIPTION
               "The Cell Delay Variation Tolerance for the VC"
       ::= { cktEntry 152 }

cktNdcEnable OBJECT-TYPE
       SYNTAX  INTEGER {
               off (1),
               on (2)
               }
       ACCESS  read-write
       STATUS  mandatory
       DESCRIPTION
               "Enable/disable Network Data Collection for a PVC on an
               IOM. The number of NDC-monitored PVCs is limited as
               follows:
               30 per DS3 interface, 90 per OC3 interface, and 360 per
               OC12 interface."
       ::= { cktEntry 153 }

cktInterworkingFrToAtmCLP OBJECT-TYPE
       SYNTAX  INTEGER {
               clp0  (0),
               clp1  (1),
                fr-de (2)
               }
       ACCESS  read-write
       STATUS  mandatory
       DESCRIPTION
               "CLP mapping for Frame Relay To ATM Service Interworking."
       ::= { cktEntry 154 }

cktInterworkingFrToAtmDe OBJECT-TYPE
       SYNTAX  INTEGER {
               de0     (0),
               de1     (1),
                atm-clp (2)
       }
       ACCESS  read-write
       STATUS  mandatory
       DESCRIPTION
               "DE bit mapping for ATM To Frame Relay  Service
               Interworking."
       ::= { cktEntry 155 }

cktNrtsCLP1 OBJECT-TYPE
       SYNTAX  INTEGER {
               no (1),
               yes (2)
       }
       ACCESS  read-write
       STATUS  mandatory
       DESCRIPTION
               "Whether to use the discard threshold as the CLP1
               Discard threshold upon reception of a cell by the NRTS
               processor in the forward direction."
       ::= { cktEntry 156 }

cktNrtsDiscardClp0 OBJECT-TYPE
       SYNTAX Counter
       ACCESS read-only
       STATUS mandatory
       DESCRIPTION
               "The number of the CLP0 cells received and discarded by the
               NRTS processor."
       ::= { cktEntry 157 }
```

```
cktNrtsDiscardClp1 OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The number of the CLP1 cells received and discarded by the
                NRTS processor."
        ::= { cktEntry 158 }

cktMPEnableAMF OBJECT-TYPE
        SYNTAX INTEGER {
        disableAMF (1),
        enableAMF (2)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Whether the Multilink PPP PVC Attributes Modification
                Feature is enabled."
        ::= { cktEntry 159 }

cktMPEligible OBJECT-TYPE
        SYNTAX INTEGER {
                isMPEligible (1),
                isNotMPEligible (2)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Whether this circuit is eligible to be a member of a
                Multilink PPP bundle."
        ::= { cktEntry 160 }

cktMPForcedCaller OBJECT-TYPE
        SYNTAX INTEGER {
                isMPForcedCaller (1),
                isNotMPForcedCaller (2)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Whether this end of the circuit has been forced to be
                caller to support Multilink PPP PVC Attributes
                Modification Feature."
        ::= { cktEntry 161 }

cktMPForcedCallee OBJECT-TYPE
        SYNTAX INTEGER {
                isMPForcedCallee (1),
                isNotMPForcedCallee (2)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Whether this end of the circuit has been forced to be
                callee to support Multilink PPP PVC Attributes
                Modification Feature."
        ::= { cktEntry 162 }

cktFrameSize OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The average frame size (measured in number of bytes) of
                the traffic on the circuit. If not configured, it is set
                to 280 (bytes) by default."
        ::= { cktEntry 163 }

cktRFrameSize    OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The average frame size (measured in number of bytes) of
                the traffic on the circuit on the reverse direction. If
                not configured, it is set to 280 (bytes) by default."
        ::= { cktEntry 164 }

cktRNrtsCLP1 OBJECT-TYPE
        SYNTAX  INTEGER {
                no (1),
                yes (2)
        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Whether to use the discard threshold as the CLP1
                Discard threshold upon reception of a cell by the NRTS
                processor in the reverse direction."
        ::= { cktEntry 165 }
```

The cktLeafTable

```
cktLeafTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF CktLeafEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A table containing information about leaves of multipoint
                circuits."
        ::= { ckt 2 }

cktLeafEntry OBJECT-TYPE
        SYNTAX  CktLeafEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "The objects relevant to a single leaf in a multipoint
                circuit."
        INDEX { cktLeafSrcIfIndex, cktLeafSrcDlci, cktLeafEndpointIndex }
        ::= { cktLeafTable 1 }

CktLeafEntry ::=
        SEQUENCE {
                cktLeafSrcIfIndex
                    Index,
                cktLeafSrcDlci
                    INTEGER,
                cktLeafEndpointIndex
                    INTEGER,
                cktLeafCreationTime
                    TimeTicks,
                cktLeafEgressIfIndex
                    Index,
                cktLeafEgressDlci
                    INTEGER,
                cktLeafDestNodeId
                    INTEGER,
                cktLeafDestIfIndex
                    Index,
                cktLeafDestDlci
                    INTEGER,
                cktLeafSvcCallingParty
                    OCTET STRING,
                cktLeafSvcCalledParty
                    OCTET STRING,
                cktLeafAdminStatus
                    INTEGER,
                cktLeafVcState
                    INTEGER,
                cktLeafOperStatus
                    INTEGER,
                cktLeafDceState
                    INTEGER,
                cktLeafDteStatus
                    INTEGER,
                cktLeafDteState
                    INTEGER,
                cktLeafVcPtr
                    OCTET STRING,
                cktLeafHopCnt
                    INTEGER,
                cktLeafPath
                    OCTET STRING,
                cktLeafFailReason
                    INTEGER,
                cktLeafFailNode
                    INTEGER,
                cktLeafFailPort
                    INTEGER,
                cktLeafHelloCounter
                    INTEGER,
                cktLeafHelloAckCounter
                    INTEGER,
                cktLeafAtmVPI
                    INTEGER,
                cktLeafAtmVCI
                    INTEGER,
                cktLeafType
                    INTEGER,
                cktLeafAtmInCells
                    Counter,
                cktLeafAtmOutCells
                    Counter,
                cktLeafAtmInDiscardedClp0Cells
                    Counter,
                cktLeafAtmInDiscardedClp1Cells
                    Counter,
                cktLeafAtmInPassedClp0Cells
                    Counter,
                cktLeafAtmInPassedClp1Cells
                    Counter,
                cktLeafAtmInTaggedCells
                    Counter,
                cktLeafAtmOutClp0Cells
                    Counter,
                cktLeafAtmOutClp1Cells
                    Counter,
                cktLeafOutPort
                    INTEGER,
                cktLeafOutVc
                    INTEGER,
                cktLeafRVc
                    INTEGER,
                cktLeafEntryType
                    INTEGER,
                cktLeafDiagStr
                    OCTET STRING,
```

```
                cktLeafEgressEndpointIndex
                    INTEGER
        }

cktLeafSrcIfIndex OBJECT-TYPE
        SYNTAX  Index
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The interface ID at this node's ingress port for the
                corresponding multipoint circuit."
        ::= { cktLeafEntry 1 }

cktLeafSrcDlci OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The DLCI, if a frame relay circuit, or concatenated VPI
                and VCI, if an ATM circuit, at this node's ingress port
                for the corresponding multipoint circuit."
        ::= { cktLeafEntry 2 }

cktLeafEndpointIndex OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "An identifier unique to this leaf for the corresponding
                multipoint circuit."
        ::= { cktLeafEntry 3 }

cktLeafCreationTime OBJECT-TYPE
        SYNTAX  TimeTicks
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The value of sysUpTime when this leaf was created for the
                corresponding multipoint circuit."
        ::= { cktLeafEntry 4 }

cktLeafEgressIfIndex OBJECT-TYPE
        SYNTAX  Index
        ACCESS  read-only
        STATUS  mandatory

        DESCRIPTION
                "The interface ID at this node's egress port for this
                leaf's segment of the corresponding multipoint circuit."
        ::= { cktLeafEntry 5 }

cktLeafEgressDlci OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The DLCI, if a frame relay circuit, or concatenated VPI
                and VCI, if an ATM circuit, at this node's egress port for
                this leaf's segment of the corresponding multipoint
                circuit."
        ::= { cktLeafEntry 6 }

cktLeafDestNodeId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The node ID of the node where this leaf terminates."
        ::= { cktLeafEntry 7 }

cktLeafDestIfIndex OBJECT-TYPE
        SYNTAX  Index
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The interface ID at the egress port of the destination
                node for this leaf, if this is a PVC."
        ::= { cktLeafEntry 8 }

cktLeafDestDlci OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The DLCI, if a frame relay circuit, or concatenated VPI
                and VCI, if an ATM circuit, at this node's destination
                port for the corresponding multipoint circuit."
        ::= { cktLeafEntry 9 }

cktLeafSvcCallingParty OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The calling party number for this leaf, if this is an
                SVC."
         ::= { cktLeafEntry 10 }

cktLeafSvcCalledParty OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The called party number for this leaf, if this is an SVC."
         ::= { cktLeafEntry 11 }
```

```
cktLeafAdminStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                configured (1),-- this leaf corresponds to a PVC
                dynamic (2),    -- this leaf corresponds to an SVC
                invalid (3)     -- this leaf shall be deleted
                }
        ACCESS  read-write
        STATUS      mandatory
        DESCRIPTION
                "The desired state for the leaf entry."
         ::= { cktLeafEntry 12 }

cktLeafVcState OBJECT-TYPE
        SYNTAX  INTEGER {
                    invalid (0),
                    inactive (1),
                    retry (2),
                    calling (3),
                    wcbdeact(4),
                    wcbdelete(5),
                    active (6),
                    svcall (7),
                    svclr (8),
                    backedup (9),
                    wcbbkdp (10),
                    wcbreact (11),
                    slowretry (12)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The current state of the leaf PVC segment in the Cascade
                network."
        ::= { cktLeafEntry 13 }

cktLeafOperStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                invalid (0),
                inactive (1),
                active (2)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The current operational status of the entire leaf PVC."
        ::= { cktLeafEntry 14 }

cktLeafDceState OBJECT-TYPE
        SYNTAX  INTEGER {
                    invalid (0),
                    inactive (1),
                    active (2)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The DCE state of the circuit."
        ::= { cktLeafEntry 15 }

cktLeafDteStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                    invalid (0),
                    inactive (1),
                    active (2)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The DTE status of the circuit."
        ::= { cktLeafEntry 16 }

cktLeafDteState OBJECT-TYPE
        SYNTAX  INTEGER {
                    invalid (0),
                    inactive (1),
                    active (2)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The DTE state of the circuit."
        ::= { cktLeafEntry 17 }

cktLeafVcPtr OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "An 8-byte Octect string indicating the vc pointer."
        ::= { cktLeafEntry 18 }

cktLeafHopCnt OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The count of hops along the circuit path. (Max is 16)"
        ::= { cktLeafEntry 19 }
```

```
cktLeafPath OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The circuit path consisting a sequence of outbound
                interface indexes at nodes along the established circuit.
                The format is
                interface:interface:interface......"
        ::= { cktLeafEntry 20 }

cktLeafFailReason OBJECT-TYPE
        SYNTAX  INTEGER {
                    none (0),
                    admindown (1),
                    novcbuff (2),
                    nobw (3),
                    noroute (4),
                    timeout (5),
                    nopdubuff (6),
                    nodest (7),
                    trkrnr (8),
                    trkdown (9),
                    balancereroute (10),
                    dead(11),
                    defpathreroute(12),
                    nidown(13),
                    otherpvcsegdown(14),
                    otherpvcsegrnr(15),
                    usingaltpathwarning(16),
                    iopdown(17),
                    numsgbuffer(18),
                    noport(19),
                    misconfig(20),
                    svcsetupfail(21),
                    srcbackedup(22),
                    srcunknown(23),
                    dstunknown(24),
                    bkpdlcicollision(25),
                    oldrevinpath(26),
                    smdsmgmttrunk(27),
                    nevercalled(28),
                    bothendptbackup(29),
                    pvcroutemgttrunk(30),
                    nomultipointparent(31),
                    pvcroutefail(32),
                    novpivci(33)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The reason for the PVC establishment failure."
        ::= { cktLeafEntry 21 }

cktLeafFailNode OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The node which causes the PVC failure."
        ::= { cktLeafEntry 22 }

cktLeafFailPort OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The port on the fail node which causes the PVC failure."
        ::= { cktLeafEntry 23 }

cktLeafHelloCounter OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Number of PVC hello pdu frames received in the VC entry
                of the called side."
        ::= { cktLeafEntry 24 }

cktLeafHelloAckCounter OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Number of PVC hello Ack pdu frames received in the VC
                entry of the calling side."
        ::= { cktLeafEntry 25 }

cktLeafAtmVPI OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "VPI value in the ATM cell header"
        ::= { cktLeafEntry 26 }

cktLeafAtmVCI OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "VCI value in the ATM cell header"
        ::= { cktLeafEntry 27 }
```

```
cktLeafType OBJECT-TYPE
        SYNTAX  INTEGER  {
                    pvc (1),
                    svc (2)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "1 if a permanent virtual circuit; 2 if a switched virtual
                circuit."
        ::= { cktLeafEntry 28 }

cktLeafAtmInCells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION "The Number of ATM cells received on a VCC."
        ::= { cktLeafEntry 29 }

cktLeafAtmOutCells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION "The Number of ATM cells transmitted on a VCC."
        ::= { cktLeafEntry 30 }

cktLeafAtmInDiscardedClp0Cells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION "The Number of ATM CLP 0 cells received and discarded
        on a VCC."
        ::= { cktLeafEntry 31 }

cktLeafAtmInDiscardedClp1Cells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The Number of ATM CLP 1 cells received and discarded
                on a VCC."
        ::= { cktLeafEntry 32 }

cktLeafAtmInPassedClp0Cells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The Number of ATM CLP 0 cells received and passed UPC on
                a VCC."
        ::= { cktLeafEntry 33 }

cktLeafAtmInPassedClp1Cells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The Number of ATM CLP 1 cells received and passed UPC on
                a VCC."
        ::= { cktLeafEntry 34 }

cktLeafAtmInTaggedCells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The Number of ATM cells received and tagged on a VCC."
        ::= { cktLeafEntry 35 }

cktLeafAtmOutClp0Cells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                The Number of ATM CLP 0 cells transmitted on a VCC."
        ::= { cktLeafEntry 36 }

cktLeafAtmOutClp1Cells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The Number of ATM CLP 1 cells transmitted on a VCC."
        ::= { cktLeafEntry 37 }

cktLeafOutPort OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The outgoing port number for the adjacent VC entry in\
                this switch."
        ::= { cktLeafEntry 38 }

cktLeafOutVc OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The adjacent VC entry corresponding to this circuit
                accross the bus."
        ::= { cktLeafEntry 39 }
```

```
cktLeafRVc OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The adjacent VC entry corresponding to this circuit
                across the trunk."
        ::= { cktLeafEntry 40 }

cktLeafEntryType OBJECT-TYPE
        SYNTAX  INTEGER {
                    fr-user(1),
                    as-trunk(2),
                    on-trunk(3),
                    lmi(4),
                    multicast(5),
                    mgmt(6),
                    smds(7),
                    split-multicast(8),
                    control(9),
                    atm-user(10),
                    atm-leaf(11)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The internal type of circuit entry allocated by the
                circuit manager."
        ::= { cktLeafEntry 41 }

cktLeafDiagStr OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Internal diagnostic information."
        ::= { cktLeafEntry 42 }

cktLeafEgressEndpointIndex OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "An identifier unique to this leaf at this node's egress
                port for this leaf's segment of the corresponding
                multipoint circuit."
        ::= { cktLeafEntry 43 }
```

Circuit Table for Managing SMDS Routes

```
cktSmdsRtTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF CktSmdsRtEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A table containing information about destination, VC,
                hops and routes."
        ::= { ckt 3 }

cktSmdsRtEntry OBJECT-TYPE
        SYNTAX  CktSmdsRtEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "The information regarding routes from a node to all its
                neighbors."
        INDEX { cktSmdsRemoteNode }
        ::= { cktSmdsRtTable 1 }

CktSmdsRtEntry ::=
        SEQUENCE {
                cktSmdsRemoteNode
                    INTEGER,
                cktSmdsHopCnt
                    INTEGER,
                cktSmdsRoute
                    OCTET STRING,
                cktSmdsLocalPort
                    INTEGER,
                cktSmdsRemotePort
                    INTEGER,
                cktSmdsVcState
                    INTEGER
        }

cktSmdsRemoteNode OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The nodeId of the node to which a rought is sought."
        ::= { cktSmdsRtEntry 1 }

cktSmdsHopCnt OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of hops to reach the desired destination node."
        ::= { cktSmdsRtEntry 2 }
```

```
cktSmdsRoute OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The out-going Interface Id of each of the nodes in the
                path."
        ::= { cktSmdsRtEntry 3 }

cktSmdsLocalPort OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The out-going local portId."
        ::= { cktSmdsRtEntry 4 }

cktSmdsRemotePort OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The remote portId of the destination nodeId."
        ::= { cktSmdsRtEntry 5 }

cktSmdsVcState OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The state of the circuit."
        ::= { cktSmdsRtEntry 6 }
```

A Circuit Table for Network Data Collection per GR-1248

```
cktNdcTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF CktNdcEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A list of Network Data Collection statistics for
                a PVC."
        ::= { ckt 4 }

cktNdcEntry OBJECT-TYPE
        SYNTAX  CktNdcEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A record of Network Data Collection statistics for
                a PVC."
        INDEX { cktNdcIfIndex, cktNdcSrcDlci, cktNdcHistIndex }
        ::= { cktNdcTable 1 }

CktNdcEntry ::=
        SEQUENCE {
                cktNdcIfIndex
                    Index,
                cktNdcSrcDlci
                    INTEGER,
                cktNdcHistIndex
                    INTEGER,
                cktNdcTimeStamp
                    INTEGER,
                cktNdcInClp01Cells
                    Counter,
                cktNdcOutClp01Cells
                    Counter,
                cktNdcInDiscardClp0Cells
                    Counter,
                cktNdcInDiscardClp01Cells
                    Counter,
                cktNdcInTaggedCells
                    Counter,
                cktNdcInDiscardClp0CellThresh
                    Counter,
                cktNdcInDiscardClp01CellThresh
                    Counter
                }

cktNdcIfIndex OBJECT-TYPE
        SYNTAX Index
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The ifIndex value of the corresponding ifEntry."
        ::= { cktNdcEntry 1 }

cktNdcSrcDlci OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The DLCI used as the key for the ATM PVC.
                The VPI (most significant 16 bits) and VCI (least
                significant 16 bits) are concatenated to form this value."
        ::= { cktNdcEntry 2 }
```

```
cktNdcHistIndex OBJECT-TYPE
        SYNTAX  INTEGER {
                current (1),
                history1 (2),
                history2 (3)
        }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "An index of the Network Data Collection history."
        ::= { cktNdcEntry 3 }

cktNdcTimeStamp OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "For the current counts (cktNDChistIndex of 1), time
                elapsed in the current 15-minute NDC collection interval.
                For the history counts (cktNDChistIndex of 2 to 3),
                timestamp at the end of 15-minute NDC collection interval.
                Resolution is 1 second."
        ::= { cktNdcEntry 4 }

cktNdcInClp01Cells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A count of user+OAM CLP=0+1 cells incoming on a circuit,
                received within the Network Data Collection 15-minute
                interval."
        ::= { cktNdcEntry 5 }

cktNdcOutClp01Cells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A count of user+OAM CLP=0+1 cells outgoing on a circuit,
                transmitted within the Network Data Collection 15-minute
                interval."
        ::= { cktNdcEntry 6 }

cktNdcInDiscardClp0Cells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A count of user+OAM CLP=0 cells incoming on a circuit,
                discarded due to UPC/NPC policing within the Network Data
                Collection 15-minute interval."
        ::= { cktNdcEntry 7 }

cktNdcInDiscardClp01Cells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A count of user+OAM CLP=0+1 cells incoming on a circuit,
                discarded due to UPC/NPC policing within the Network Data
                Collection 15-minute interval."
        ::= { cktNdcEntry 8 }

cktNdcInTaggedCells OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A count of user+OAM CLP=0 cells incoming on a circuit,
                tagged as CLP=1 due to UPC/NPC policing within the Network
                Data Collection 15-minute interval."
        ::= { cktNdcEntry 9 }

cktNdcInDiscardClp0CellThresh OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "A count of user+OAM CLP=0 cells incoming on a circuit,
                discarded due to UPC/NPC policing within the Network Data
                Collection current 15-minute interval. An alarm is issued
                once in the current interval upon crossing of that
                threshold provided
                that the threshold is greater than 0 (default)."
        ::= { cktNdcEntry 10 }

cktNdcInDiscardClp01CellThresh OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "A threshold count of user+OAM CLP=0+1 cells incoming on a
                circuit, discarded due to UPC/NPC policing within the
                Network Data Collection current 15-minute interval. An
                alarm is issued once in the current interval upon crossing
                of that threshold provided that the threshold is greater
                than 0 (default)."
        ::= { cktNdcEntry 11 }
```

the Cascade DS1 Configuration Table

```
cascds1ConfigTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF Cascds1ConfigEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "The DS1 Configuration table."
        ::= { cascds1 1 }

cascds1ConfigEntry OBJECT-TYPE
        SYNTAX  Cascds1ConfigEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "An entry in the DS1 Configuration table."
        INDEX   { cascds1SlotId, cascds1PortId }
        ::= { cascds1ConfigTable 1 }
```

```
Cascds1ConfigEntry ::=
        SEQUENCE {
                cascds1SlotId
                    INTEGER,
                cascds1PortId
                    INTEGER,
                cascds1TimeElapsed
                    INTEGER,
                cascds1ValidIntervals
                    INTEGER,
                cascds1LineType
                    INTEGER,
                cascds1LineCoding
                    INTEGER,
                cascds1SendCode
                    INTEGER,
                cascds1CircuitIdentifier
                    DisplayString,
                cascds1LoopbackConfig
                    INTEGER,
                cascds1LineStatus
                    INTEGER,
                cascds1SignalMode
                    INTEGER,
                cascds1TransmitClockSource
                    INTEGER,
                cascds1Fdl
                    INTEGER,
                cascds1FdlVersion
                    INTEGER
                }

cascds1SlotId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The slot number of the corresponding DS1."
        ::= { cascds1ConfigEntry 1 }

cascds1PortId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The port number of the corresponding DS1
                on the board."
        ::= { cascds1ConfigEntry 2 }
```

```
cascds1TimeElapsed OBJECT-TYPE
        SYNTAX  INTEGER (0..899)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of seconds that have elapsed  since
                the  beginning of the current error-measurement
                period."
        ::= { cascds1ConfigEntry 3 }

cascds1ValidIntervals OBJECT-TYPE
        SYNTAX  INTEGER (0..96)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of  previous  intervals  for  which
                valid data was collected.  The value will be 96
                unless the interface was brought on-line within
                the last 24 hours, in which case the value will
                be the number of complete 15  minute  intervals
                the since interface has been online."
        ::= { cascds1ConfigEntry 4 }

cascds1LineType OBJECT-TYPE
        SYNTAX  INTEGER {
                other(1),
                cascds1ESF(2),
                cascds1D4(3),
                cascds1E1(4),
                cascds1E1-CRC(5),
                cascds1E1-MF(6),
                cascds1E1-CRC-MF(7)
        }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This variable indicates  the  variety  of  DS1
                Line  implementing  this  circuit.  The type of
                circuit affects the number of bits  per  second
                that  the circuit can reasonably carry, as well
                as the interpretation of the  usage  and  error
                statistics.  The values, in sequence, describe:

                TITLE:          SPECIFICATION:
                cascds1ESF      Extended SuperFrame DS1
                cascds1D4       AT&T D4 format DS1
                cascds1E1       CCITT Recommendation G.704
                                (Table 4a)
                cascds1E1-CRC   CCITT Recommendation G.704
                                (Table 4b)
                cascdsE1-MF      G.704 (Table 4a) with TS16
                                multiframing enabled
                cascds1E1-CRC-MF G.704 (Table 4b) with TS16
                                multiframing enabled"

        ::= { cascds1ConfigEntry 5 }

cascds1LineCoding OBJECT-TYPE
        SYNTAX  INTEGER {
                cascds1JBZS (1),
                cascds1B8ZS (2),
                cascds1HDB3 (3),
                cascds1ZBTSI (4),
                cascds1AMI (5),
                other(6)
        }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This variable describes the  variety  of  Zero
                Code  Suppression  used  on  the link, which in
                turn affects a number of its characteristics.
                cascds1JBZS refers the Jammed  Bit  Zero  Suppres-
                sion,  in  which  the  AT&T specification of at
                least one pulse every 8 bit periods is literal-
                ly  implemented  by forcing a pulse in bit 8 of
                each channel.  Thus, only seven bits per  chan-
                nel, or 1.344 Mbps, is available for data.
                cascds1B8ZS refers to the use of a specified  pat-
                tern  of  normal  bitsand  bipolar violations
                which are used to replace a sequence  of  eight
                zero bits.
                ANSI Clear Channels may use cascds1ZBTSI, or  Zero
                Byte Time Slot Interchange.
                E1 links, with or without CRC, use cascds1HDB3or
                cascds1AMI.
                cascds1AMI refers to a mode wherein nozero  code
                suppression  is  present  and the line encoding
                does not solve the problem directly.   In  this
                application, the higher layer must provide data
                which meets or exceeds the  pulse  density  re-
                quirements, such as inverting HDLC data."
        ::= { cascds1ConfigEntry 6 }
```

```
cascds1SendCode OBJECT-TYPE
        SYNTAX  INTEGER {
                cascds1SendNoCode(1),
                cascds1SendLineCode(2),
                cascds1SendPayloadCode(3),
                cascds1SendResetCode(4),
                cascds1SendQRS(5),
                cascds1Send511Pattern(6),
                cascds1Send3in24Pattern(7),
                cascds1SendOtherTestPattern(8)
        }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This variable indicates what type of  code  is
                being sent across the DS1 interface by the dev-
                ice.  The values mean:

                cascds1SendNoCode
                    sending looped or normal data

                cascds1SendLineCode
                    sending a request for a line loopback

                cascds1SendPayloadCode
                    sending a request for a payload loopback

                cascds1SendResetCode
                    sending a loopback termination request

                cascds1SendQRS
                    sending a Quasi-Random Signal  (QRS)  test
                    pattern

                cascds1Send511Pattern
                    sending a 511 bit fixed test pattern

                cascds1Send3in24Pattern
                    sending a fixed test pattern of 3 bits set
                    in 24

                cascds1SendOtherTestPattern
                    sending a test patternother  than  those
                    described by this object"
        ::= { cascds1ConfigEntry 7 }
```

```
cascds1CircuitIdentifier OBJECT-TYPE
        SYNTAX  DisplayString (SIZE (0..255))
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This  variable   contains   the   transmission
                vendor's circuit identifier, for the purpose of
                facilitating troubleshooting."
        ::= { cascds1ConfigEntry 8 }

cascds1LoopbackConfig OBJECT-TYPE
        SYNTAX  INTEGER {
                cascds1NoLoop(1),
                cascds1PayloadLoop(2),
                cascds1LineLoop(3),
                cascds1OtherLoop(4)
        }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This variable represents the  loopback  confi-
                guration of the DS1 interface.Agents support-
                ing read/write access should return badValue in
                response to a requested loopback state that the
                interface does not support.  The values mean:

                cascds1NoLoop
                    Not in the loopback state.  A device  that
                    is not capable of performing a loopback on
                    the interface shall always return thisas
                    it's value.

                cascds1PayloadLoop
                    The received signal at this  interfaceis
                    looped     through  the device. Typically the
                    received signal is  looped  back  for  re-
                    transmission  after  it has passed through
                    the device's framing function.

                cascds1LineLoop
                    The received signal at this interface does
                    not  go  through the device (minimum pene-
                    tration) but is looped back out.

                cascds1OtherLoop
                    Loopbacks that are not defined here."
        ::= { cascds1ConfigEntry 9 }
```

```
cascds1LineStatus OBJECT-TYPE
        SYNTAX  INTEGER (1..8191)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This variable indicates the Line Status of the
                interface.   It contains loopback, failure, re-
                ceived 'alarm'  and transmitted 'alarm'  infor-
                mation.

                The cascds1LineStatus is a bitmap  represented  as  a  sum,
                therefore, it can represent multiple failures (alarms) and
                a LoopbackState simultaneously.

                cascds1NoAlarm should be set if and only if no other flag
                is set.

                If the cascds1LoopbackState bit is set, the loopback  in
                effect can be determined from the cascds1LoopbackConfig
                object.

                The various bit positions are:
                1     cascds1NoAlarm            No Alarm Present
                2     cascds1RcvFarEndLOF       Far end LOF (a.k.a., Yellow
                                                Alarm)
                4     cascds1XmtFarEndLOF       Near end sending LOF
                                                Indication
                8     cascds1RcvAIS             Far end sending AIS
                16    cascds1XmtAIS             Near end sending AIS
                32    cascds1LossOfFrame        Near end LOF (a.k.a., Red
                                                Alarm)
                64    cascds1LossOfSignal       Near end Loss Of Signal
                128   cascds1LoopbackState      Near end is looped
                256   cascds1T16AIS             E1 TS16 AIS
                512   cascds1RcvFarEndLOMF      Far End Sending TS16 LOMF
                1024  cascds1XmtFarEndLOMF      Near End Sending TS16 LOMF
                2048  cascds1RcvTestCode        Near End detects a test code
                4096  cascds1OtherFailure       any line status not defined
                                                here"
        ::= { cascds1ConfigEntry 10 }
```

```
cascds1SignalMode OBJECT-TYPE
        SYNTAX  INTEGER {
                none (1),
                robbedBit (2),
                bitOriented (3),
                messageOriented (4)
        }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "'none' indicates that no bits are reserved for
                signaling on this channel.

                'robbedBit' indicates that T1 Robbed Bit  Sig-
                naling is in use.

                'bitOriented' indicates that E1 Channel  Asso-
                ciated Signaling is in use.

                'messageOriented' indicates that Common  Chan-
                nel Signaling is in use either on channel 16 of
                an E1 link or channel 24 of a T1."
        ::= { cascds1ConfigEntry 11 }

cascds1TransmitClockSource OBJECT-TYPE
        SYNTAX  INTEGER {
                loopTiming (1),
                localTiming (2),
                throughTiming (3)
        }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The source of Tranmit Clock.

                'loopTiming' indicates that the recovered  re-
                ceive clock is used as the transmit clock.

                'localTiming' indicates  that  a  local  clock
                source is used.

                'throughTiming' indicates that  recovered  re-
                ceive  clock  from another interface is used as
                the transmit clock."
        ::= { cascds1ConfigEntry 12 }
```

```
cascds1Fdl OBJECT-TYPE
        SYNTAX  INTEGER {
                other(1),
                cascds1Ansi-T1-403(2),
                cascds1Att-54016(4),
                cascds1Fdl-none(8)
        }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This bitmap describes the use of  the  facili-
                ties data link, and is the sum of the capabili-
                ties:

                'other' indicates that a protocol  other  than
                one following is used.

                'cascds1Ansi-T1-403' refers to the  FDL  exchange
                recommended by ANSI.

                'cascds1Att-54016' refers to ESF FDL exchanges.

                'cascds1Fdl-none' indicates that the device  does
                not use the FDL."
        ::= { cascds1ConfigEntry 13 }

cascds1FdlVersion OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The version of the Dallas FDL firmware"
        ::= { cascds1ConfigEntry 14 }
```

the DS1 Current Table

```
-- The DS1 current table contains various statistics being
-- collected for the current 15 minute interval.

cascds1CurrentTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF Cascds1CurrentEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "The DS1 Current table."
        ::= { cascds1 2 }

cascds1CurrentEntry OBJECT-TYPE
        SYNTAX  Cascds1CurrentEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "An entry in the DS1 Current table."
        INDEX   { cascds1CurrentSlotId, cascds1CurrentPortId }
        ::= { cascds1CurrentTable 1 }

Cascds1CurrentEntry ::=
        SEQUENCE {
                cascds1CurrentSlotId
                    INTEGER,
                cascds1CurrentPortId
                    INTEGER,
                cascds1CurrentESs
                    Gauge,
                cascds1CurrentSESs
                    Gauge,
                cascds1CurrentSEFSs
                    Gauge,
                cascds1CurrentUASs
                    Gauge,
                cascds1CurrentCSSs
                    Gauge,
                cascds1CurrentBESs
                    Gauge
        }

cascds1CurrentSlotId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The slot number of the corresponding DS1."
        ::= { cascds1CurrentEntry 1 }
```

```
cascds1CurrentPortId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The port number of the corresponding DS1
                on the board."
        ::= { cascds1CurrentEntry 2 }

cascds1CurrentESs OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of Errored Seconds, encountered  by
                a DS1 interface in the current 15 minute inter-
                val."
        ::= { cascds1CurrentEntry 3 }

cascds1CurrentSESs OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of Severely Errored Seconds encoun-
                tered  by  a  DS1  interface  in the current 15
                minute interval."
        ::= { cascds1CurrentEntry 4 }

cascds1CurrentSEFSs OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of Severely Errored Framing Seconds
                encountered  by  a DS1 interface in the current
                15 minute interval."
        ::= { cascds1CurrentEntry 5 }

cascds1CurrentUASs OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of Unavailable Seconds  encountered
                by a DS1 interface in the current 15 minute in-
                terval."
        ::= { cascds1CurrentEntry 6 }

cascds1CurrentCSSs OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of Controlled Slip Seconds  encoun-
                tered  by  a  DS1  interface  in the current 15
                minute interval."
        ::= { cascds1CurrentEntry 7 }

cascds1CurrentBESs OBJECT-TYPE
        SYNTAX Gauge
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The number of Bursty  Errored  Seconds  (BESs)
                encountered  by  a DS1 interface in the current
                15 minute interval."
        ::= {cascds1CurrentEntry 8 }
```

the DS1 Interval

```
-- The DS1 Interval Table contains various statistics
-- collected by each DS1 Interface over the previous 24 hours of
-- operation.  The past 24 hours are broken into 96 completed
-- 15 minute intervals.

cascds1IntervalTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF Cascds1IntervalEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "The DS1 Interval table."
        ::= { cascds1 3 }

cascds1IntervalEntry OBJECT-TYPE
        SYNTAX  Cascds1IntervalEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "An entry in the DS1 Interval table."
        INDEX   { cascds1IntervalSlotId, cascds1IntervalPortId,
                cascds1IntervalNumber }
        ::= { cascds1IntervalTable 1 }
```

```
Cascds1IntervalEntry ::=
        SEQUENCE {
                cascds1IntervalSlotId
                    INTEGER,
                cascds1IntervalPortId
                    INTEGER,
                cascds1IntervalNumber
                    INTEGER,
                cascds1IntervalESs
                    Gauge,
                cascds1IntervalSESs
                    Gauge,
                cascds1IntervalSEFSs
                    Gauge,
                cascds1IntervalUASs
                    Gauge,
                cascds1IntervalCSSs
                    Gauge,
                cascds1IntervalBESs
                    Gauge
        }

cascds1IntervalSlotId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The slot number of the corresponding DS1."
        ::= { cascds1IntervalEntry 1 }

cascds1IntervalPortId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The port number of the corresponding DS1
                on the board."
        ::= { cascds1IntervalEntry 2 }

cascds1IntervalNumber OBJECT-TYPE
        SYNTAX  INTEGER (1..96)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A number between 1 and 96, where 1 is the most
                recently completed 15 minute interval and 96 is
                the least recently completed 15 minutes  inter-
                val   (assuming   thatall  96  intervals  are
                valid)."
        ::= { cascds1IntervalEntry 3 }

cascds1IntervalESs OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of Errored Seconds encountered by a
                DS1  interface in one of the previous 96, indi-
                vidual 15 minute, intervals."
        ::= { cascds1IntervalEntry 4 }

cascds1IntervalSESs OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of Severely Errored Seconds encoun-
                tered by a DS1 interface in one of the previous
                96, individual 15 minute, intervals."
        ::= { cascds1IntervalEntry 5 }

cascds1IntervalSEFSs OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of Severely Errored Framing Seconds
                encountered  by  a  DS1 interface in one of the
                previous 96, individual 15 minute, intervals."
        ::= { cascds1IntervalEntry 6 }

cascds1IntervalUASs OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of Unavailable Seconds  encountered
                by  a  DS1 interface in one of the previous 96,
                individual 15 minute, intervals."
        ::= { cascds1IntervalEntry 7 }

cascds1IntervalCSSs OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of Controlled Slip Seconds  encoun-
                tered by a DS1 interface in one of the previous
                96, individual 15 minute, intervals."
        ::= { cascds1IntervalEntry 8 }
```

```
cascds1IntervalBESs OBJECT-TYPE
        SYNTAX Gauge
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The number of Bursty  Errored  Seconds  (BESs)
                encountered  by  a  DS1 interface in one of the
                previous 96, individual 15 minute, intervals."
        ::= {cascds1IntervalEntry 9 }
-- the DS1 Total
-- The DS1 Total Table contains the cumulative sum of the
-- various statistics for the 24 hour period preceding the
-- current interval.

cascds1TotalTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF Cascds1TotalEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "The DS1 Total table.  24 hour interval."
        ::= { cascds1 4 }

cascds1TotalEntry OBJECT-TYPE
        SYNTAX  Cascds1TotalEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "An entry in the DS1 Total table."
        INDEX   { cascds1TotalSlotId, cascds1TotalPortId }
        ::= { cascds1TotalTable 1 }

Cascds1TotalEntry ::=
        SEQUENCE {
                cascds1TotalSlotId
                    INTEGER,
                cascds1TotalPortId
                    INTEGER,
                cascds1TotalESs
                    Gauge,
                cascds1TotalSESs
                    Gauge,
                cascds1TotalSEFSs
                    Gauge,
                cascds1TotalUASs
                    Gauge,
                cascds1TotalCSSs
                    Gauge,
                cascds1TotalBESs
                    Gauge
        }
```

```
cascds1TotalSlotId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The slot number of the corresponding DS1."
        ::= { cascds1TotalEntry 1 }

cascds1TotalPortId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The port number of the corresponding DS1
                on the board."
        ::= { cascds1TotalEntry 2 }

cascds1TotalESs OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of Errored Seconds encountered by a
                DS1 interface in the previous 24 hour interval"
        ::= { cascds1TotalEntry 3 }

cascds1TotalSESs OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of Severely Errored Seconds encoun-
                tered  by  a  DS1  interface in the previous 24
                hour interval."
        ::= { cascds1TotalEntry 4 }

cascds1TotalSEFSs OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of Severely Errored Framing Seconds
                encountered  by a DS1 interface in the previous
                24 hour interval."
        ::= { cascds1TotalEntry 5 }
```

```
cascds1TotalUASs OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of Unavailable Seconds  encountered
                by  a DS1 interface in the previous 24 hour in-
                terval."
        ::= { cascds1TotalEntry 6 }

cascds1TotalCSSs OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of Controlled Slip Seconds  encoun-
                tered  by  a  DS1  interface in the previous 24
                hour interval."
        ::= { cascds1TotalEntry 7 }

cascds1TotalBESs OBJECT-TYPE
        SYNTAX Gauge
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The number of Bursty  Errored  Seconds  (BESs)
                encountered  by a DS1 interface in the previous
                24 hour interval."
        ::= { cascds1TotalEntry 8 }
```

SMDS address : An SMDS address can be a local individual address

```
-- assigned to a DXI/SNI, a local group address defined in the STDX,
-- an individual address which is not assigned to any DXI/SNI in this
-- STDX but is a member of an individual address screen, or can be
-- a group address which is not defined in the STDX but is a member
-- of a group address screen.

smdsaddrTable OBJECT-TYPE
        SYNTAX SEQUENCE OF SmdsaddrEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "A list of smds address entries."
        ::= { smdsaddr 1 }

smdsaddrEntry OBJECT-TYPE
        SYNTAX SmdsaddrEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "An smds address and its associated information."
        INDEX { smdsaddrAddr }
        ::= { smdsaddrTable 1 }

SmdsaddrEntry ::=
        SEQUENCE {
                smdsaddrAddr
                    OCTET STRING,
                smdsaddrType
                    INTEGER,
                smdsaddrId
                    INTEGER,
                smdsaddrIf
                    INTEGER,
                smdsaddrParentGrpMap
                    OCTET STRING,
                smdsaddrParentScrnMap
                    OCTET STRING,
                smdsaddrMemberAddrMap
                    OCTET STRING,
                smdsaddrAdminStatus
                    INTEGER,
                smdsaddrSlot
                    INTEGER,
                smdsaddrSsiIfNum
                    INTEGER
                }

smdsaddrAddr OBJECT-TYPE
        SYNTAX OCTET STRING
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "SMDS address. The 4 most significant bits are the address
                type : 1100 for an individual address, 1110 for a group
                address. The following 4 bits are 0001. The following 5
                bytes are the 10 digits number in BCD format. The
                following 16 bits are padded with 1's"
        ::= { smdsaddrEntry 1 }
```

```
smdsaddrType OBJECT-TYPE
        SYNTAX INTEGER {
                local-individual-address(1),
                local-group-address(2),
                non-local-individual-address(3),
                non-local-group-address(4),
                distributed-individual-address(5),
                ssi-feeder-address(6)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "SMDS address internal type."
        ::= { smdsaddrEntry 2 }

smdsaddrId OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "smds address internal ID.
                Range from 1 to 192 for local individual addresses.
                Range from 193 to 704 for Alien individual addresses.
                Range from 1 to 64 for local group addresses.
                Range from 65 to 576 for alien group addresses."
        ::= { smdsaddrEntry 3 }

smdsaddrIf OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The interface to which this address is assigned."
        ::= { smdsaddrEntry 4 }

smdsaddrParentGrpMap OBJECT-TYPE
        SYNTAX OCTET STRING
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The parent group bit map. The bit position of each bit
                in this bit map represent a parent group address ID. The
                most significant bit is corresponding to group address ID
                1. This MIB object applies to a local individual address
                only."
        ::= { smdsaddrEntry 5 }

smdsaddrParentScrnMap OBJECT-TYPE
        SYNTAX OCTET STRING
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "This MIB object not applicable to switch software versions
                03.99.00 and above.
                The parent screen bit map. The bit position of each bit
                in this bit map represent a parent screen ID. The most
                significant bit is corresponding to screen ID 1."
        ::= { smdsaddrEntry 6 }

smdsaddrMemberAddrMap OBJECT-TYPE
        SYNTAX OCTET STRING
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The member individual address bit map. For set request,
                the first byte is the operation : 1 for deleting, 2 for
                adding. The following bytes are the bit
                map. The bit position of each bit in this bit map represent
                a member local individual address ID. The most significant
                bit is corresponding to local individual address ID 1. For
                get response, The whole string is the bit map."
        ::= { smdsaddrEntry 7 }

smdsaddrAdminStatus OBJECT-TYPE
        SYNTAX INTEGER {
                invalid(0),
                down(1),
                up(2)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "administration status"
        ::= { smdsaddrEntry 8 }

smdsaddrSlot OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The Slot in which this address is assigned."
        ::= { smdsaddrEntry 9 }

smdsaddrSsiIfNum OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The SSI to which this address is associated."
        ::= { smdsaddrEntry 10 }
```

ISDN Addr Group

```
-- The ISDN addr group is comprised of two tables:
-- 1) indexed based on interface id, this table contains the lport E.164
address
-- and the lport type - b-channel or sw56
-- 2) indexed based on interface id, this table contains valid caller ids

isdnAddrTable OBJECT-TYPE
        SYNTAX SEQUENCE OF IsdnAddrEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "A list of E.164 address entries."
        ::= { isdnaddr 1 }

isdnAddrEntry OBJECT-TYPE
        SYNTAX IsdnAddrEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "Information about a single local E.164 address in the
                table. Setting this variable to 0 removes the entry."
        INDEX { isdnAddrIf }
        ::= { isdnAddrTable 1 }

IsdnAddrEntry ::=
        SEQUENCE {
                isdnAddrIf
                    INTEGER,
                isdnAddr
                    OCTET STRING,
                isdnChanType
                    INTEGER
                }

isdnAddrIf OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The interface id of the logical port in question"
        ::= { isdnAddrEntry 1 }

isdnAddr OBJECT-TYPE
        SYNTAX OCTET STRING
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "ISDN address. The actual E164 number. For example,
                ISDN address 15086922600 is stored as string 15086922600.
                Setting this value to a null string removes this entry."
        ::= { isdnAddrEntry 2 }

isdnChanType OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                " type of logical port 0 => isdn b-channel,
                1 => sw56 "
        ::= { isdnAddrEntry 3}

isdnCallerIDTable OBJECT-TYPE
        SYNTAX SEQUENCE OF IsdnCallerIDEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "A list of valid caller ids "
        ::= { isdnaddr 2 }

isdnCallerIDEntry OBJECT-TYPE
        SYNTAX IsdnCallerIDEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "Information about a single valid caller id."
                INDEX { isdnCallerIDIf, isdnCallerIDAddr }
        ::= { isdnCallerIDTable 1 }

IsdnCallerIDEntry ::=
        SEQUENCE {
                isdnCallerIDIf
                    INTEGER,
                isdnCallerIDAddr
                    OCTET STRING,
                isdnCallerAdminStatus
                    INTEGER
                }
```

```
isdnCallerIDIf OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The interface id of the logical port in question"
        ::= { isdnCallerIDEntry 1 }

isdnCallerIDAddr OBJECT-TYPE
        SYNTAX OCTET STRING
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "ISDN address. The actual E164 number. For example,
                ISDN address 15086922600 is stored as string 15086922600."
        ::= { isdnCallerIDEntry 2 }

isdnCallerAdminStatus OBJECT-TYPE
        SYNTAX INTEGER {
                remove (0),
                add (1)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "administration status"
        ::= { isdnCallerIDEntry 3 }
```

DVC group

```
--      The DVC group consists of one or more tables:
--      1) DVC ckt group table
--      In addition to this, protocol specific tables can be added here if
        seen fit.
--      Although this is not aesthetically pleasing, it seems the most
        practical
--      way to allow for protocol specific differences without getting
        embroiled
--      in too much academic discussion.

--  DVC CKT GRP THINGS:

dvcCktGrpTable  OBJECT-TYPE
        SYNTAX SEQUENCE OF DvcCktGrpEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "A list of DVC Circuit Group entries defined in a card"
        ::= { dvccktgrp 1 }

dvcCktGrpEntry OBJECT-TYPE
        SYNTAX DvcCktGrpEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "A DVC Circuit Group entry contains information about a DVC
                Circuit Group"
        INDEX { dvcCktGrpNodeId, dvcCktGrpSlotId, dvcCktGrpId }
        ::= { dvcCktGrpTable 1 }

DvcCktGrpEntry ::=
        SEQUENCE {
                dvcCktGrpNodeId
                    INTEGER,
                dvcCktGrpSlotId
                    INTEGER,
                dvcCktGrpId
                    INTEGER,
                dvcCktGrpAdminStatus
                    INTEGER,
                dvcCktGrpMaxDvcs
                    INTEGER,
                dvcCktGrpActiveDvcCount
                    INTEGER,
                dvcCktGrpDialedE164Addr
                    OCTET STRING,
                dvcCktGrpPPPOption
                    INTEGER,
                dvcCktGrpAuthDomainId
                    INTEGER,
                dvcCktGrpBaseIpAddr
                    IpAddress,
                dvcCktGrpIngressLportProtocol
                    INTEGER,
                dvcCktGrpAuthAdminStatus
                    INTEGER,
                dvcCktGrpEgressBeginDlci
                    INTEGER,
                dvcCktGrpEgressNodeId
                    INTEGER,
                dvcCktGrpEgressSlotId
                    INTEGER,
                dvcCktGrpEgressIfNum
                    INTEGER,
                dvcCktGrpPriority
                    INTEGER,
                dvcCktGrpCir
                    INTEGER,
                dvcCktGrpBc
                    INTEGER,
                dvcCktGrpBe
                    INTEGER,
                dvcCktGrpOde
                    INTEGER,
```

```
                dvcCktGrpChanType
                    INTEGER,
                dvcCktXlatFlag
                    INTEGER,
                dvcCktAtmInCells
                    Counter,
                dvcCktAtmOutCells
                    Counter,
                dvcCktAtmInDiscardedClp0Cells
                    Counter,
                dvcCktAtmInDiscardedClp1Cells
                    Counter,
                dvcCktAtmPCR
                    INTEGER,
                dvcCktAtmSCR
                    INTEGER,
                dvcCktAtmMBS
                    INTEGER,
                dvcCktGrpActiveDvcs
                    OCTET STRING,
                dvcCktGrpValidateCaller
                    INTEGER
                }

dvcCktGrpNodeId  OBJECT-TYPE
        SYNTAX   INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Node Id  in which this variable resides"
        ::= {dvcCktGrpEntry 1}

dvcCktGrpSlotId  OBJECT-TYPE
        SYNTAX   INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Slot Id  in which this DVC Circuit Group resides"
        ::= {dvcCktGrpEntry 2}

dvcCktGrpId   OBJECT-TYPE
        SYNTAX   INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Id of the DVC Ckt Grp; unique within a map"
        ::= {dvcCktGrpEntry 3}

dvcCktGrpAdminStatus  OBJECT-TYPE
        SYNTAX   INTEGER {
                add (1),
                idle(2),
                remove(3)
    }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Admin Status of the DVC Circuit Group"
        ::= {dvcCktGrpEntry 4}

dvcCktGrpMaxDvcs  OBJECT-TYPE
        SYNTAX   INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Maximum number of live DVCs allowed for this Circuit
                Group"
        ::= {dvcCktGrpEntry 5}

dvcCktGrpActiveDvcCount  OBJECT-TYPE
        SYNTAX   INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Actual number of live DVCs in this Circuit Group"
        ::= {dvcCktGrpEntry 6}

dvcCktGrpDialedE164Addr  OBJECT-TYPE
        SYNTAX   OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Dialed E164 address of the DVC Circuit Group"
        ::= {dvcCktGrpEntry 7}

dvcCktGrpPPPOption  OBJECT-TYPE
        SYNTAX INTEGER {
                pap-only (1),
                chap-only (2),
                pap-and-chap (3)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "PPP authentication options."
        ::= {dvcCktGrpEntry 8}
```

```
dvcCktGrpAuthDomainId  OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Authentication Domain ID for this lport."
        ::= {dvcCktGrpEntry 9}

dvcCktGrpBaseIpAddr  OBJECT-TYPE
        SYNTAX   IpAddress
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Base Ip address for this circuit group"
        ::= {dvcCktGrpEntry 10}

dvcCktGrpIngressLportProtocol  OBJECT-TYPE
        SYNTAX   INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Ingress lport protocol for this circuit group"
        ::= {dvcCktGrpEntry 11}

dvcCktGrpAuthAdminStatus OBJECT-TYPE
        SYNTAX INTEGER {
                auth-enabled (1),
                auth_disabled (2)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Authentication enabled for this port, yes or no."
        ::= {dvcCktGrpEntry 12}

dvcCktGrpEgressBeginDlci OBJECT-TYPE
        SYNTAX   INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Egress beginning DLCI for this circuit group: if either
                this is 0, implies use any. This variable is a overloaded
                with a 16 bit VPI/VCI for ATM support"
        ::= {dvcCktGrpEntry 13}

dvcCktGrpEgressNodeId  OBJECT-TYPE
        SYNTAX   INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Egress node id for this circuit group"
        ::= {dvcCktGrpEntry 14}

dvcCktGrpEgressSlotId  OBJECT-TYPE
        SYNTAX   INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Egress slot id for this circuit group"
        ::= {dvcCktGrpEntry 15}

dvcCktGrpEgressIfNum  OBJECT-TYPE
        SYNTAX   INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Beginning Egress Interface Number for this circuit group"
        ::= {dvcCktGrpEntry 16}

dvcCktGrpPriority  OBJECT-TYPE
        SYNTAX   INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Circuit priority (1 through 3) for each DVC in this
                circuit group"
        ::= {dvcCktGrpEntry 17}

dvcCktGrpCir   OBJECT-TYPE
        SYNTAX   INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "CIR for each DVC in this circuit group"
        ::= {dvcCktGrpEntry 18}

dvcCktGrpBc   OBJECT-TYPE
        SYNTAX   INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Bc for each DVC in this circuit group"
        ::= {dvcCktGrpEntry 19}

dvcCktGrpBe   OBJECT-TYPE
        SYNTAX   INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Be for each DVC in this circuit group"
        ::= {dvcCktGrpEntry 20}
```

```
dvcCktGrpOde   OBJECT-TYPE
        SYNTAX INTEGER {
               off (0),
               on (1)
               }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
               "ODE for each DVC in this circuit group"
        ::= {dvcCktGrpEntry 21}

dvcCktGrpChanType OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
               " type of logical port 0 => isdn b-channel,
               1 => isdn sw56 "
        ::= { dvcCktGrpEntry 22}

dvcCktXlatFlag OBJECT-TYPE
        SYNTAX  INTEGER
               {
               no-translation (0),
               rfc1483 (1)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
               "0 if RFC1490 or no translation; 1 if RFC1483
               translation."
        ::= { dvcCktGrpEntry 23 }

dvcCktAtmInCells OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION "The Number of ATM cells received on a VC (VPC or
                    VCC)."
        ::= { dvcCktGrpEntry 24 }

dvcCktAtmOutCells OBJECT-TYPE
        SYNTAX Counter
        ACCESS  read-only
        STATUS mandatory
        DESCRIPTION "The Number of ATM cells transmitted on a VC (VPC or
               VCC).."
        ::= { dvcCktGrpEntry 25 }

dvcCktAtmInDiscardedClp0Cells OBJECT-TYPE
        SYNTAX Counter
        ACCESS  read-only
        STATUS mandatory
        DESCRIPTION "The Number of ATM CLP0 cells received and discarded
               on a VC (VPC or VCC).."
        ::= { dvcCktGrpEntry 26 }

dvcCktAtmInDiscardedClp1Cells OBJECT-TYPE
        SYNTAX Counter
        ACCESS  read-only
        STATUS mandatory
        DESCRIPTION "The Number of ATM CLP1 cells received and discarded
               on a VC (VPC or VCC)."
        ::= { dvcCktGrpEntry 27 }

dvcCktAtmPCR OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
               "The Peak Cell Rate  measured in cells/second at which
               cells are transmitted for this circuit."
        ::= { dvcCktGrpEntry 28 }

dvcCktAtmSCR OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
               "The Sustainable Cell Rate is the average transmission rate
               in cells per second for this circuit. It must be less than
               or or equal to the Peak Cell Rate."
        ::= { dvcCktGrpEntry 29 }

dvcCktAtmMBS OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
               "The Maximum Burst size determines the maximum number of
               cells that can be transmitted at the peak cell rate."
        ::= { dvcCktGrpEntry 30 }
```

```
dvcCktGrpActiveDvcs  OBJECT-TYPE
        SYNTAX   OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This is a (6 + m *  12) octet BINARY string where:
                m is the number of live DVCs in this circuit group ( m >=
                0).The first 2 bytes is the DVC Circuit Group id.
                The second 2 bytes is the length of this string including
                the circuit group id.
                Byte #5 is the length of data for each DVC.
                Byte #6 is number of live DVCs described in this string.
                Byte #7 through Byte #18 describe the first DVC,
                Byte #19 through Byte #30 describe the second DVC and so
                on..
                Each Live DVC is described in the following format:
                Ingress Node, Ingress Ifnum, Ingress DLCI, Egress DLCI
                Note that Byte 5 is used to accommodate future expansion to
                include other data. Hence, always add to the end of this
                tuple, not to the middle.
                "
        ::= { dvcCktGrpEntry 31 }

dvcCktGrpValidateCaller  OBJECT-TYPE
        SYNTAX INTEGER {
                validateCaller (1),
                dontValidateCaller (2)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Does this cktgrp require callerid screening?"
        ::= { dvcCktGrpEntry 32 }

--    Protocol specific MIB variables related to DVC go here
--   -----------------------------------------------------

isdnAuthenCallerIDTable OBJECT-TYPE
        SYNTAX SEQUENCE OF IsdnAuthenCallerIDEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "A list of Authentic caller ids and the DVC circuit groups
                they are allowed to access"
        ::= { dvcprotocustom 1 }

isdnAuthenCallerIDEntry OBJECT-TYPE
        SYNTAX IsdnAuthenCallerIDEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "Information about a single authentic caller id."
        INDEX { isdnSlotId, isdnDvcCktGrpId, isdnAuthenCallerIDAddr }
        ::= { isdnAuthenCallerIDTable 1 }

IsdnAuthenCallerIDEntry ::=
        SEQUENCE {
                isdnSlotId
                    INTEGER,
                isdnDvcCktGrpId
                    INTEGER,
                isdnAuthenCallerIDAddr
                    OCTET STRING,
                isdnAuthenCallerAdminStatus
                    INTEGER
                }

isdnSlotId OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "slot id"
        ::= { isdnAuthenCallerIDEntry 1 }

isdnDvcCktGrpId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The DVC Circuit Group Id that this caller is allowed to
                access"
        ::= { isdnAuthenCallerIDEntry 2 }

isdnAuthenCallerIDAddr OBJECT-TYPE
        SYNTAX  OCTET STRING (SIZE(1..16))
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "ISDN address. The actual E164 number. For example,
                ISDN address 15086922600 is stored as null-terminated
                string 15086922600.This string has a maximum size of
                16 bytes."
        ::= { isdnAuthenCallerIDEntry 3 }
```

```
isdnAuthenCallerAdminStatus OBJECT-TYPE
        SYNTAX INTEGER {
                add (1),
                remove (2)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "administration status"
        ::= { isdnAuthenCallerIDEntry 4 }
```

The Service Name Binding Group

```
--  The variables that are relevant to a Service Name Binding Table

namebindingTable OBJECT-TYPE
        SYNTAX SEQUENCE OF NamebindingEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "A table of name bindings to a set of logical ports."
        ::= { namebinding 1 }

namebindingEntry OBJECT-TYPE
        SYNTAX NamebindingEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "The name binding entry contains objects relative to a
                name binding."
        INDEX { nameType, nameName, namePrimary }
        ::= { namebindingTable 1 }

NamebindingEntry ::=
        SEQUENCE
                {
                nameType
                    INTEGER,
                nameName
                    OCTET STRING,
                namePrimary
                    INTEGER,
                nameIfIndex
                    INTEGER,
                nameNodeId
                    INTEGER,
                nameStatus
                    INTEGER
                }

nameType OBJECT-TYPE
        SYNTAX INTEGER {
                uninniladdr(1), -- this is a logical address for frame
                relay
                e164(2), -- this is an E.164 address
                nsap(3), -- this is an NSAP address
                sni(4)  -- this is an SNI SMDS address
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The name type."
        ::= { namebindingEntry 1 }

nameName OBJECT-TYPE
        SYNTAX OCTET STRING
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "A name of the type indicated by nameType."
        ::= { namebindingEntry 2 }

namePrimary OBJECT-TYPE
        SYNTAX INTEGER {
                primary(1), -- a primary backup
                backup(2) -- a backup binding
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The current binding type for this entry."
        ::= { namebindingEntry 3 }

nameIfIndex OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The interface number of the logical port for this
                binding."
        ::= { namebindingEntry 4 }

nameNodeId OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The identifier number of the node for this binding."
        ::= { namebindingEntry 5 }
nameStatus OBJECT-TYPE
        SYNTAX INTEGER {
                active(1), -- binding is active
                invalid(2) -- binding is invalid and is deleted
        }
```

```
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The status of the binding."
        ::= { namebindingEntry 6 }
```

The SVC Address Group

```
--  The tables that are relevant to managing SVC addresses and prefixes
--  in a Cascade network.
```

SVC Node Prefix Table

```
svcNodePrefixTable OBJECT-TYPE
        SYNTAX SEQUENCE OF SvcNodePrefixEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "A table of prefixes associated with this node."
        ::= { svcaddress 1 }

svcNodePrefixEntry OBJECT-TYPE
        SYNTAX SvcNodePrefixEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "The node prefix entry contains objects relevant to SVC
                prefixes associated with the node.  Note that the index
                variable, svcNodePrefixPrefix is a variable length octet
                string and as such is encoded with the octet string length
                per RFC1212, section 4.1.6."
        INDEX { svcNodePrefixPrefix }
        ::= { svcNodePrefixTable 1 }

SvcNodePrefixEntry ::=
        SEQUENCE {
                svcNodePrefixPrefix
                    OCTET STRING,
                svcNodePrefixNumBits
                    INTEGER,
                svcNodePrefixNmbPlan
                    INTEGER,
                svcNodePrefixAdminStatus
                    INTEGER,
                svcNodePrefixAttributes
                     INTEGER,
                svcNodePrefixCugStat
                    OCTET STRING
                }

svcNodePrefixPrefix OBJECT-TYPE
        SYNTAX OCTET STRING (SIZE(1..20))
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "A prefix associated with this node.  E.164 prefixes are
                coded as 1-15 ASCII octets with no leading padding
                required.  ATM endsystem prefixes are coded as 1-20 binary
                octets.  Unused bits in the last octet must be set to 0."
        ::= { svcNodePrefixEntry 1 }

svcNodePrefixNumBits OBJECT-TYPE
        SYNTAX INTEGER (1..160)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The number of valid bits associated with this entry's
                prefix object. By default, this value will be 8 times the
                prefix object's octet string length.  This value must be
                consistent with the number of octets specified in the node
                prefix."
        ::= { svcNodePrefixEntry 2 }

svcNodePrefixNmbPlan OBJECT-TYPE
        SYNTAX INTEGER {
                e164 (1),
                atm-endsystem (2),
                unknown (4)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The numbering plan corresponding to this entry's prefix
                object."
        ::= { svcNodePrefixEntry 3 }
svcNodePrefixAdminStatus OBJECT-TYPE
        SYNTAX INTEGER {
                configured (1),  -- this entry has been configured by NMS
```

```
                invalid (2)  -- this entry shall be deleted
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The status of this entry."
        ::= { svcNodePrefixEntry 4 }

svcNodePrefixAttributes OBJECT-TYPE
        SYNTAX INTEGER (0..32767)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object supports a bitwise encoding of attributes and
                 capabilities associated with this node prefix.

                The defined values are as follows:
                2 - Route determination
                4 - Source address validation
                8 - ILMI address registration"
        ::= { svcNodePrefixEntry 5 }

svcNodePrefixCugStat OBJECT-TYPE
        SYNTAX OCTET STRING
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "This object encodes the CUG status for this
                particular row.  The encoding scheme is as follows:
                E - Error
                OA - Outgoing Access
                IA - Incoming Access
                # - CUG Identifier
                O - Outgoing Calls Barred
                I - Incoming Calls Barred
                P - Preferential CUG

        Order is important.  The following output:

        E OA | 1 I | 10 P

        means Error, Outgoing Access, CUG Identifier 1,
        Incoming Access, CUG Identifier 10, Preferential"
        ::= { svcNodePrefixEntry 6 }
```

SVC Prefix Table

```
svcPrefixTable OBJECT-TYPE
        SYNTAX SEQUENCE OF SvcPrefixEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "A table of SVC address prefixes associated with ports on
                this node."
        ::= { svcaddress 2 }

svcPrefixEntry OBJECT-TYPE
        SYNTAX SvcPrefixEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "The prefix entry contains objects relevant to SVC prefixes
                associated with this port.  Note that the index variable,
                svcPrefixPrefix is a variable length octet string and as
                such is encoded with the octet string length per RFC1212,
                section 4.1.6."
        INDEX { svcPrefixIfIndex, svcPrefixPrefix }
        ::= { svcPrefixTable 1 }

SvcPrefixEntry ::=
        SEQUENCE {
                svcPrefixIfIndex
                    Index,
                svcPrefixPrefix
                    OCTET STRING,
                svcPrefixNumBits
                    INTEGER,
                svcPrefixNmbPlan
                    INTEGER,
                svcPrefixAdminCost
                    INTEGER,
                svcPrefixLocalGatewayAddress
                    OCTET STRING,
                svcPrefixLocalGatewayNmbPlan
                    INTEGER,
                svcPrefixRemoteGatewayAddress
                    OCTET STRING,
                svcPrefixRemoteGatewayNmbPlan
                    INTEGER,
                svcPrefixAdminStatus
                    INTEGER,
                svcPrefixAttributes
                    INTEGER,
                svcPrefixCugStat
                    OCTET STRING
                }
```

```
svcPrefixIfIndex OBJECT-TYPE
        SYNTAX Index
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The interface value of the corresponding MIB-II ifEntry."
        ::= { svcPrefixEntry 1 }

svcPrefixPrefix OBJECT-TYPE
        SYNTAX OCTET STRING (SIZE(1..20))
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "A prefix associated with this port.  E.164 prefixes are
                coded as 1-15 ASCII octets with no leading padding
                required.  ATM endsystem prefixes are coded as 1-20 binary
                octets.  Unused bits in the last octet must be set to 0.

                For ATM DCE ports, only, atm-endsystem prefixes with
                length 104 bits (13 octets) and all E.164 prefixes are
                eligible for ILMI address registration."
        ::= { svcPrefixEntry 2 }

svcPrefixNumBits OBJECT-TYPE
        SYNTAX INTEGER (0..160)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The number of valid bits associated with this entry's
                prefix object.By default, this value will be 8 times the
                prefix object's octet string length.  This value must be
                consistent with the number of octets specified in the
                prefix.  The value 0 presents a special case and may
                only be set when the prefix, itself, is a single octet of
                value 0.  A 0-length prefix on this port signifies a
                default route to the switch's routing function."
        ::= { svcPrefixEntry 3 }

svcPrefixNmbPlan OBJECT-TYPE
        SYNTAX INTEGER {
                e164 (1),
                atm-endsystem (2),
                unknown (4)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The numbering plan corresponding to this entry's prefix
                object."
        ::= { svcPrefixEntry 4 }

svcPrefixAdminCost OBJECT-TYPE
        SYNTAX INTEGER (0..65535)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The administrative cost associated with this prefix."
        ::= { svcPrefixEntry 5 }

svcPrefixLocalGatewayAddress OBJECT-TYPE
        SYNTAX OCTET STRING (SIZE(0..20))
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object is only relevant for ports connecting this
                network to another network and is used to replace the
                calling party number when egress address translation is
                configured to the appropriate mode.
                E.164 addresses are coded as 1-15 ASCII octets  Atm-
                endsystem addresses are coded as 20 octet binary
                addresses.  A 0 length octet string will invalidate this
                object."
        ::= { svcPrefixEntry 6 }

svcPrefixLocalGatewayNmbPlan OBJECT-TYPE
        SYNTAX INTEGER {
                e164 (1),
                atm-endsystem (2),
                unknown (4)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The numbering plan corresponding to this entry's local
                gateway address object."
        ::= { svcPrefixEntry 7 }

svcPrefixRemoteGatewayAddress OBJECT-TYPE
        SYNTAX OCTET STRING (SIZE(0..20))
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object is only relevant for ports connecting this
                network to another network and is used to replace the
                calling party number when egress address translation is
                configured to the appropriate mode. E.164 addresses are
                coded as 1-15 ASCII octets  Atm-endsystem addresses are
                coded as 20 octet binary addresses.  A 0 length octet
                string will invalidate this object."
        ::= { svcPrefixEntry 8 }
```

```
svcPrefixRemoteGatewayNmbPlan OBJECT-TYPE
        SYNTAX INTEGER {
                e164 (1),
                atm-endsystem (2),
                unknown (4)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The numbering plan corresponding to this entry's remote
                gateway address object."
        ::= { svcPrefixEntry 9 }

svcPrefixAdminStatus OBJECT-TYPE
        SYNTAX INTEGER {
                configured (1),  -- this entry has been configured by NMS
                invalid (2),  -- this entry shall be deleted
                dynamic (3)  -- this entry created via peer device (ATM
                UNI DTE, only)
                }
                ACCESS read-write
                STATUS mandatory
                DESCRIPTION
                    "The status of this entry."
                ::= { svcPrefixEntry 10 }

svcPrefixAttributes OBJECT-TYPE
        SYNTAX INTEGER (0..32767)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object supports a bitwise encoding of attributes and
                  capabilities associated with this prefix.

                The defined values are as follows:
                2 - Route determination
                4 - Source address validation
                8 - ILMI address registration
                32 - CUG Termination
                64 - CUG Future Use"
        ::= { svcPrefixEntry 11 }

svcPrefixCugStat OBJECT-TYPE
        SYNTAX OCTET STRING
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "This object encodes the CUG status for this
                particular row.  The encoding scheme is as follows:

                E - Error
                OA - Outgoing Access
                IA - Incoming Access
                # - CUG Identifier
                O - Outgoing Calls Barred
                I - Incoming Calls Barred
                P - Preferential CUG

                Order is important.  The following output:

                E OA | 1 I | 10 P

                means Error, Outgoing Access, CUG Identifier 1,
                Incoming Access, CUG Identifier 10, Preferential"
        ::= { svcPrefixEntry 12 }
```

The SMDS Address Group

```
--  An SMDS address can be a local individual address assigned to a
--  DXI/SNI, a local group address defined in the STDX, an individual
--  address which is not assigned to any DXI/SNI in this STDX, but is
--  a member of an individual address screen, or can be a group
--  address which is not defined in the STDX but is a member of a group
--  address screen.

smdsaddrTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF SmdsaddrEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A list of smds adress entries."
        ::= { smdsaddr 1 }

smdsaddrEntry OBJECT-TYPE
        SYNTAX  SmdsaddrEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "An smds address and its associated information."
        INDEX   { smdsaddrAddr }
        ::= { smdsaddrTable 1 }

SmdsaddrEntry ::=
        SEQUENCE {
                smdsaddrAddr
                    OCTET STRING,
                smdsaddrType
                    INTEGER,
                smdsaddrId
                    INTEGER,
                smdsaddrIf
                    INTEGER,
                smdsaddrParentGrpMap
                    OCTET STRING,
                smdsaddrParentScrnMap
                    OCTET STRING,
                smdsaddrMemberAddrMap
                    OCTET STRING,
                smdsaddrAdminStatus
                    INTEGER,
                smdsaddrSlot
                    INTEGER,
                smdsaddrSsiIfNum
                    INTEGER
                }

smdsaddrAddr OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "SMDS address. The 4 most significant bits are the address
                type: 1100 for an individual address, 1110 for a group
                address. The following 4 bits are 0001. The following 5
                bytes are the 10 digits number in BCD format. The
                following 16 bits are padded with 1's"
        ::= { smdsaddrEntry 1 }

smdsaddrType OBJECT-TYPE
        SYNTAX  INTEGER {
                local-individual-address(1),
                local-group-address(2),
                non-local-individual-address(3),
                non-local-group-address(4)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "SMDS address internal type."
        ::= { smdsaddrEntry 2 }

smdsaddrId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "smds address internal ID.
                Range from 1 to 192 for local individual addresses.
                Range from 193 to 704 for Alien individual addresses.
                Range from 1 to 64 for local group addresses.
                Range from 65 to 576 for alien group addresses."
        ::= { smdsaddrEntry 3 }

smdsaddrIf OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The interface to which this address is assigned."
        ::= { smdsaddrEntry 4 }
```

```
smdsaddrParentGrpMap OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The parent group bit map. The bit position of each bit in
                this bit map represent a parent group address ID. The most
                significant bit is corresponding to group address ID 1.
                This MIB object applies to a local individual address
                only."
        ::= { smdsaddrEntry 5 }

smdsaddrParentScrnMap OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This MIB object not applicable to switch software versions
                03.99.00 and above.
                The parent screen bit map. The bit position of each bit
                in this bit map represent a parent screen ID. The most
                significant bit is corresponding to screen ID 1."
        ::= { smdsaddrEntry 6 }

smdsaddrMemberAddrMap OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The member individual address bit map. For set request,
                the first byte is the operation : 1 for deleting, 2 for
                adding. The following bytes are the bit map. The bit
                position of each bit in this bit map represents a member
                local individual address ID. The most significant bit is
                corresponding to local individual address ID 1. For get
                response, the whole string is the bit map."
        ::= { smdsaddrEntry 7 }

smdsaddrAdminStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                invalid(0),
                down(1),
                up(2)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "administration status"
        ::= { smdsaddrEntry 8 }

smdsaddrSlot OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The Slot in which this address is assigned."
        ::= { smdsaddrEntry 9 }

smdsaddrSsiIfNum OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The SSI to which this address is associated."
        ::= { smdsaddrEntry 10 }
```

The ISDN Address Group

```
--  The ISDN addr group is comprised of two tables:
--  1) indexed based on interface id, this table contains the lport E.164
--  address and the lport type - b-channel or sw56
--  2) indexed based on interface id, this table contains valid caller ids

isdnAddrTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF IsdnAddrEntry
        ACCESS  not-accessible
        STATUS  mandatory

        DESCRIPTION
                "A list of E.164 address entries."
        ::= { isdnaddr 1 }

isdnAddrEntry OBJECT-TYPE
        SYNTAX  IsdnAddrEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "Information about a single local E.164 address in the
                table. Setting this varible to 0 removes the entry."
        INDEX   { isdnAddrIf }
        ::= { isdnAddrTable 1 }

IsdnAddrEntry ::=
        SEQUENCE {
                isdnAddrIf
                    INTEGER,
                isdnAddr
                    OCTET STRING,
                isdnChanType
                    INTEGER
                }
```

```
isdnAddrIf OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The interface id of the logical port in question"
        ::= { isdnAddrEntry 1 }

isdnAddr OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "ISDN address. The actual E164 number. For example,
                ISDN address 15086922600 is stored as string 15086922600.
                Setting this value to a null string removes this entry."
        ::= { isdnAddrEntry 2 }

isdnChanType OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Type of logical port 0 => isdn b-channel, 1 => sw56."
        ::= { isdnAddrEntry 3}

isdnCallerIDTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF IsdnCallerIDEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A list of valid caller ids."
        ::= { isdnaddr 2 }

isdnCallerIDEntry OBJECT-TYPE
        SYNTAX  IsdnCallerIDEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "Information about a single valid caller id."
        INDEX   { isdnCallerIDIf, isdnCallerIDAddr }
        ::= { isdnCallerIDTable 1 }

IsdnCallerIDEntry ::=
        SEQUENCE {
                isdnCallerIDIf
                    INTEGER,
                isdnCallerIDAddr
                    OCTET STRING,
                isdnCallerAdminStatus
                    INTEGER
        }

isdnCallerIDIf OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The interface id of the logical port in question"
        ::= { isdnCallerIDEntry 1 }

isdnCallerIDAddr OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "ISDN address. The actual E164 number. For example,
                ISDN address 15086922600 is stored as string 15086922600."
         ::= { isdnCallerIDEntry 2 }

isdnCallerAdminStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                remove (0),
                add (1)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "administration status"
        ::= { isdnCallerIDEntry 3 }
```

The Service Name Binding Group

```
--  The variables that are relevant to a Service Name Binding Table

namebindingTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF NamebindingEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A table of name bindings to a set of logical ports."
        ::= { namebinding 1 }

namebindingEntry OBJECT-TYPE
        SYNTAX  NamebindingEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "The name binding entry contains objects relative to a
                name binding."
        INDEX { nameType, nameName, namePrimary }
        ::= { namebindingTable 1 }
```

```
NamebindingEntry ::=
        SEQUENCE {
                nameType
                INTEGER,
                nameName
                OCTET STRING,
                namePrimary
                INTEGER,
                nameIfIndex
                INTEGER,
                nameNodeId
                INTEGER,
                nameStatus
                INTEGER
        }

nameType OBJECT-TYPE
        SYNTAX  INTEGER {
                uninniladdr(1),-- this is a logical address for frame relay
                e164(2),       -- this is an E.164 address
                nsap(3),       -- this is an NSAP address
                sni(4),        -- this is an SNI SMDS address
        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The name type."
        ::= { namebindingEntry 1 }

nameName OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "A name of the type indicated by nameType."
        ::= { namebindingEntry 2 }

namePrimary OBJECT-TYPE
        SYNTAX  INTEGER {
                primary(1),-- a primary backup
                backup(2)-- a backup binding
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The current binding type for this entry."
        ::= { namebindingEntry 3 }

nameIfIndex OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The interface number of the logical port for this
                binding."
        ::= { namebindingEntry 4 }

nameNodeId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The identifier number of the node for this binding."
        ::= { namebindingEntry 5 }

nameStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                active(1),     -- binding is active
                invalid(2)     -- binding is invalid and is deleted
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The status of the binding."
        ::= { namebindingEntry 6 }
```

The SVC Address Group

```
--  The tables that are relevant to managing SVC addresses and prefixes
--  in a Cascade network.  Will eventually obsolete the cascfr.svcaddr
--  group.
```

SVC Node Prefix Table

```
svcNodePrefixTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF SvcNodePrefixEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A table of SVC address prefixes associated with this
                node."
        ::= { svcaddress 1 }
```

```
svcNodePrefixEntry OBJECT-TYPE
        SYNTAX SvcNodePrefixEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "The node prefix entry contains objects relevant to SVC
                prefixes associated with the node.  Note that the index
                variable, svcNodePrefixPrefix is a variable length octet
                string and as such is encoded with the octet string length
                per RFC1212, section 4.1.6."
        INDEX { svcNodePrefixPrefix }
        ::= { svcNodePrefixTable 1 }

SvcNodePrefixEntry ::=
        SEQUENCE {
                svcNodePrefixPrefix
                    OCTET STRING,
                svcNodePrefixNumDigits
                    INTEGER,
                svcNodePrefixNmbPlan
                    INTEGER,
                svcNodePrefixAdminStatus
                    INTEGER
        }

svcNodePrefixPrefix OBJECT-TYPE
        SYNTAX  OCTET STRING (SIZE(8..20))
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The index for this entry is a left-justified, 0-padded,
                SVC address prefix.  E.164 entries require an 8 octet
                string while SVC End System entries require a 20 octet
                string."
        ::= { svcNodePrefixEntry 1 }

svcNodePrefixNumDigits OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The number of significant BCD digits, if numbering plan is
                E.164, or the number of significant octets, if numbering
                plan is SVC End System, for this entry."
        ::= { svcNodePrefixEntry 2 }

svcNodePrefixNmbPlan OBJECT-TYPE
        SYNTAX  INTEGER {
                e164 (1),
                atmEndSystem (2),
                unknown (4)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The numbering plan used for this SVC address prefix
                entry."
        ::= { svcNodePrefixEntry 3 }

svcNodePrefixAdminStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                configured (1), -- this entry has been configured by NMS
                invalid (2)    -- this entry shall be deleted
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The status of this entry."
        ::= { svcNodePrefixEntry 4 }
```

SVC Prefix Table

```
svcPrefixTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF SvcPrefixEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A table of SVC address prefixes associated with ports on
                this node."
        ::= { svcaddress 2 }
svcPrefixEntry OBJECT-TYPE
        SYNTAX SvcPrefixEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "The prefix entry contains objects relevant to SVC prefixes
                 associated with this port.  Note that the index variable,
                 svcPrefixPrefix is a variable length octet string and as
                 such is encoded with the octet string length per RFC1212,
                 section 4.1.6."
        INDEX { svcPrefixIfIndex, svcPrefixPrefix }
        ::= { svcPrefixTable 1 }
```

```
SvcPrefixEntry ::=
        SEQUENCE {
                svcPrefixIfIndex
                    Index,
                svcPrefixPrefix
                    OCTET STRING,
                svcPrefixNumBits
                    INTEGER,
                svcPrefixNmbPlan
                    INTEGER,
                svcPrefixAdminCost
                    INTEGER,
                svcPrefixLocalGatewayAddress
                    OCTET STRING,
                svcPrefixLocalGatewayNmbPlan
                    INTEGER,
                svcPrefixRemoteGatewayAddress
                    OCTET STRING,
                svcPrefixRemoteGatewayNmbPlan
                    INTEGER,
                svcPrefixAdminStatus
                    INTEGER
        }

svcPrefixIfIndex OBJECT-TYPE
        SYNTAX Index
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The interface value of the corresponding MIB-II ifEntry."
        ::= { svcPrefixEntry 1 }

svcPrefixPrefix OBJECT-TYPE
        SYNTAX OCTET STRING (SIZE(1..20))
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "A prefix associated with this port.  E.164 prefixes are
                coded as 1-15 ASCII octets with no leading padding
                required.  ATM endsystem prefixes are coded as 1-20 binary
                octets.  Unused bits in the last
                 octet must be set to 0.

                 For ATM DCE ports, only, atm-endsystem prefixes with
                length 104 bits (13 octets) and all E.164 prefixes are
                eligible for ILMI address registration."
        ::= { svcPrefixEntry 2 }

svcPrefixNumBits OBJECT-TYPE
        SYNTAX INTEGER (0..160)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The number of valid bits associated with this entry's
                prefix object.  By default, this value will be 8 times the
                prefix object's octet string length.  This value must be
                consistent with the number of octets specified in the
                prefix.  The value 0 presents a special case and may only
                be set when the prefix, itself, is a single octet of value
                0.  A 0-length prefix on this port signifies a default
                route to the switch's routing function."
        ::= { svcPrefixEntry 3 }

svcPrefixNmbPlan OBJECT-TYPE
        SYNTAX  INTEGER {
                e164 (1),
                atmEndSystem (2),
                unknown (4)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The numbering plan used for this SVC address prefix
                entry."
        ::= { svcPrefixEntry 4 }

svcPrefixAdminCost OBJECT-TYPE
        SYNTAX INTEGER (0..65535)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object is only relevant for ports connecting this
                network to another network and is used to represent the
                administrative cost of reaching destinations containing
                this entry's prefix value."
        ::= { svcPrefixEntry 5 }

svcPrefixLocalGatewayAddress OBJECT-TYPE
        SYNTAX OCTET STRING (SIZE(0..20))
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object is only relevant for ports connecting this
                network to another network and is used to replace the
                calling party number when egress address translation is
                configured to the appropriate mode. E.164 addresses are
                coded as 1-15 ASCII octets  Atm-endsystem addresses
                are coded as 20 octet binary addresses.  A 0 length octet
                string will invalidate this object."
        ::= { svcPrefixEntry 6 }
```

```
svcPrefixLocalGatewayNmbPlan OBJECT-TYPE
        SYNTAX INTEGER{
                e164 (1),
                atm-endsystem (2),
                unknown (4)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The numbering plan corresponding to this entry's local
                gateway address object."
        ::= { svcPrefixEntry 7 }

svcPrefixRemoteGatewayAddress OBJECT-TYPE
        SYNTAX OCTET STRING (SIZE(0..20))
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object is only relevant for ports connecting this
                network to another network and is used to replace the
                calling party number when egress address translation is
                configured to the appropriate mode. E.164 addresses are
                coded as 1-15 ASCII octets  Atm-endsystem addresses
                are coded as 20 octet binary addresses.  A 0 length octet
                string will  invalidate this object."
        ::= { svcPrefixEntry 8 }

svcPrefixRemoteGatewayNmbPlan OBJECT-TYPE
        SYNTAX INTEGER{
                e164 (1),
                atm-endsystem (2),
                unknown (4)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The numbering plan corresponding to this entry's remote
                gateway address object."
        ::= { svcPrefixEntry 9 }

svcPrefixAdminStatus OBJECT-TYPE
        SYNTAX INTEGER {
                configured (1), -- this entry has been configured by NMS
                invalid (2),    -- this entry shall be deleted
                dynamic (3)     -- this entry created via peer device (ATM
                                UNI DTE, only)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The status of this entry."
        ::= { svcPrefixEntry 10 }
```

SVC Addr Table

```
svcAddrTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF SvcAddrEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A table of SVC addresses associated with with ports on
                this node."
        ::= { svcaddress 3 }

svcAddrEntry OBJECT-TYPE
        SYNTAX SvcAddrEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "The  address entry contains objects relevant to SVC
                addresses associated with this port.  Note that the index
                variable, svcAddrAddress is a variable length octet string
                and as such is encoded with the octet string length per
                RFC1212, section 4.1.6."
        INDEX { svcAddrIfIndex, svcAddrAddress }
        ::= { svcAddrTable 1 }

SvcAddrEntry ::=
        SEQUENCE {
                svcAddrIfIndex
                    Index,
                svcAddrAddress
                    OCTET STRING,
                svcAddrNmbPlan
                    INTEGER,
                svcAddrAdminStatus
                    INTEGER,
                svcAddrAdminCost
                    INTEGER,
                svcAddrAttributes
                    INTEGER,
                svcAddrCugStat
                    OCTET STRING,
                svcAddrPvcConnId
                    INTEGER
                }

svcAddrIfIndex OBJECT-TYPE
        SYNTAX Index
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The interface value of the corresponding MIB-II ifEntry."
        ::= { svcAddrEntry 1 }
```

```
svcAddrAddress OBJECT-TYPE
        SYNTAX OCTET STRING (SIZE(1..20))
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "An address associated with this port.  E.164 addresses
                are coded as 1-15 ASCII octets.  ATM-Endsystem addresses
                are coded as 20 binary octets."
        ::= { svcAddrEntry 2 }

svcAddrNmbPlan OBJECT-TYPE
        SYNTAX INTEGER {
                e164 (1),
                atm-endsystem (2),
                unknown (4)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The numbering plan corresponding to this entry's address
                object."
        ::= { svcAddrEntry 3 }

svcAddrAdminStatus OBJECT-TYPE
        SYNTAX INTEGER {
                configured (1), -- this entry has been configured by NMS
                invalid (2),    -- this entry shall be deleted
                dynamic (3)     -- this entry created by peer device (ATM
                                UNI DCE, only)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The status of this entry."
        ::= { svcAddrEntry 4 }

svcAddrAdminCost OBJECT-TYPE
        SYNTAX INTEGER (0..65535)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The administrative cost associated with this address."
        ::= { svcAddrEntry 5 }

svcAddrAttributes OBJECT-TYPE
        SYNTAX INTEGER (0..32767)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object supports a bitwise encoding of attributes and
                  capabilities associated with this address.

                The defined values are as follows:
                1 - PVC termination of SVC
                2 - Route determination
                4 - Source address validation
                16 - PVP termination of SVC
                32 - CUG Termination
                64 - CUG Future Use"
        ::= { svcAddrEntry 6 }

svcAddrCugStat OBJECT-TYPE
        SYNTAX OCTET STRING
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "This object encodes the CUG status for this
                particular row.  The encoding scheme is as follows:

                E - Error
                OA - Outgoing Access
                IA - Incoming Access
                # - CUG Identifier
                O - Outgoing Calls Barred
                I - Incoming Calls Barred
                P - Preferential CUG

                Order is important.  The following output:

                E OA | 1 I | 10 P

                means Error, Outgoing Access, CUG Identifier 1,
                Incoming Access, CUG Identifier 10, Preferential"
        ::= { svcAddrEntry 7 }
```

```
svcAddrPvcConnId OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object specifies the connection ID associated
                with this address.  It is applicable only if either
                of but not both of the PVC termination or PVP termination
                attributes are set.  It is interpreted as a 32-bit
                integer with VPI and VCI/DLCI fields as follows:

                bit 31  16 15  0
                |-----------------|--------------|
                | VPI   |   VCI/DLCI  |
                |-----------------|--------------|

                For ATM PVC/PVP termination, a VPI of 0 and VCI of 0
                indicate that the switch may select any VPI/VCI, as
                appropriate.  For PVP termination, the VCI must be
                coded as 0.

                For Frame Relay PVC termination, the VPI must be coded
                as 0.  A DLCI of 0 indicates that the switch may
                select any DLCI, as appropriate."
        ::= { svcAddrEntry 8 }
```

SVC ATM User Part Table

```
svcAtmDteUserPartTable OBJECT-TYPE
        SYNTAX SEQUENCE OF SvcAtmDteUserPartEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "A table of partial SVC addresses associated with ports on
                this node, relevant only to ATM DTE ports for use in ILMI
                address registration."
        ::= { svcaddress 4 }

svcAtmDteUserPartEntry OBJECT-TYPE
        SYNTAX SvcAtmDteUserPartEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "The user part entry contains objects relevant to partial
                SVC addresses associated with ATM DTE ports on this node."
        INDEX { svcAtmDteUserPartIfIndex, svcAtmDteUserPartUserPart }
        ::= { svcAtmDteUserPartTable 1 }

SvcAtmDteUserPartEntry ::=
        SEQUENCE {
                svcAtmDteUserPartIfIndex
                    Index,
                svcAtmDteUserPartUserPart
                    OCTET STRING,
                svcAtmDteUserPartAdminStatus
                    INTEGER
        }

svcAtmDteUserPartIfIndex OBJECT-TYPE
        SYNTAX Index
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The interface value of the corresponding MIB-II ifEntry."
        ::= { svcAtmDteUserPartEntry 1 }

svcAtmDteUserPartUserPart OBJECT-TYPE
        SYNTAX OCTET STRING (SIZE(7))
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "A partial ATM endsystem address associated with this ATM
                DTE port. It will be combined with ATM endsystem prefixes
                received from the peer DCE to form a full ATM endsystem
                address.  This object is coded as 7 binary octets."
        ::= { svcAtmDteUserPartEntry 2 }

svcAtmDteUserPartAdminStatus OBJECT-TYPE
        SYNTAX INTEGER {
                configured (1), -- this entry has been configured by NMS
                invalid (2)     -- this entry shall be deleted
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The status of this entry."
        ::= { svcAtmDteUserPartEntry 3 }
```

The SVC Management Group

```
--  The tables that are relevant to managing ATM SVC's in a Cascade
network.
```

SVC Configuration Table

```
svcConfigTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF SvcConfigEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A table of SVC configuration paramaters associated with
                logical ports. The number of entries is given by the value
                of ifNumber in MIB-II."
        ::= { svcmgt 1 }

svcConfigEntry OBJECT-TYPE
        SYNTAX  SvcConfigEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "The  SVC configuration  entry contains objects relevant
                to SVC operation on a logical port."
        INDEX   { svcConfigIfIndex }
        ::= { svcConfigTable 1 }
```

```
SvcConfigEntry ::=
        SEQUENCE {
                svcConfigIfIndex
                    Index,
                svcConfigCgPtyPresentationMode
                    INTEGER,
                svcConfigCgPtyInsertionMode
                    INTEGER,
                svcConfigCgPtyInsertionAddress
                    OCTET STRING,
                svcConfigCgPtyInsertionNmbPlan
                    INTEGER,
                svcConfigCgPtyScreenMode
                    INTEGER,
                svcConfigEgressAddrXlateMode
                    INTEGER,
                svcConfigIngressAddrXlateMode
                    INTEGER,
                svcConfigCugEnable
                    INTEGER,
                 svcConfigSecScrIngressMode
                     INTEGER,
                 svcConfigSecScrIngressDefaultScreen
                     INTEGER,
                 svcConfigSecScrEgressMode
                    INTEGER,
                svcConfigSecScrEgressDefaultScreen
                    INTEGER,
                svcConfigSvcFailureLogReset
                    INTEGER,
                svcConfigSvcFailureTrapThreshold
                    INTEGER,
                svcConfigNumSvcFailures
                    INTEGER,
                svcConfigLoadBalanceEligibilityDuration
                    INTEGER,
                svcConfigVpiVpciMappingType
                    INTEGER,
                svcConfigVpiVpciMappingOffset
                    INTEGER,
                vcConfigProxyAdminStatus
                    INTEGER,
                svcConfigProxyPSANodeId
                    INTEGER,
                svcConfigProxyPSAIfNum
                    INTEGER
                }
```

```
svcConfigIfIndex OBJECT-TYPE
        SYNTAX  Index
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The ifIndex value of the  corresponding ifEntry."
        ::= { svcConfigEntry 1 }

svcConfigCgPtyPresentationMode OBJECT-TYPE
        SYNTAX  INTEGER {
                    user (1),   -- use signalled presentation indicator
                    never (2),  -- override signalled presentation and
                                   never present
                    always (3)  -- override signalled presentation and
                                   always present
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "This object controls the circumstances under which the
                calling party number shall be presented at the egress port
                of the network."
        ::= { svcConfigEntry 2 }

svcConfigCgPtyInsertionMode OBJECT-TYPE
        SYNTAX  INTEGER {
                disabled (1),
                insert (2),-- insert when absent
                replace (3)-- insert/replace always
        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "This object determines whether a statically configured
                address shall be inserted in the calling party address IE
                for calls entering the network at this port."
        ::= { svcConfigEntry 3 }

svcConfigCgPtyInsertionAddress OBJECT-TYPE
        SYNTAX  OCTET STRING (SIZE(0..20))
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The calling party address to use in conjunction with the
                calling party insertion function.  E.164 addresses are
                coded as 1-15 ASCII octets.  Atm-endsystem addresses are
                coded as 20 binary octets. A 0 length octet string will
                NULL the address."
        ::= { svcConfigEntry 4 }

svcConfigCgPtyInsertionNmbPlan OBJECT-TYPE
        SYNTAX INTEGER {
                e164 (1),
                atm-endsystem (2),
                unknown (4)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The numbering plan corresponding to this entry's calling
                party insertion address object."
        ::= { svcConfigEntry 5 }

svcConfigCgPtyScreenMode OBJECT-TYPE
        SYNTAX  INTEGER {
                    node-prefix (1),
                    port-prefix (2),
                    node-prefix-or-port-prefix (3),
                    address (4),
                    node-prefix-or-address (5),
                    port-prefix-or-address (6),
                    node-prefix-or-port-prefix-or-address (7),
                    disabled (255)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The type of screening to perform on the calling party
                number for calls entering the network at this port.
                Screening will be conducted by performing a bit-wise (for
                prefixes) or byte-wise (for addresses) comparison of the
                calling party number with the appropriate prefix or
                address database(s), as configured."
        ::= { svcConfigEntry 6 }
```

```
svcConfigEgressAddrXlateMode OBJECT-TYPE
        SYNTAX  INTEGER {
                    disabled (1),
                    tunnel-when-called-party-matches-prefix (2),
                    replace-when-called-party-matches-prefix (3),
                    translate-e164-native-to-nsap (4),
                    translate-e164-nsap-to-native (5)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "This object determines the type of address translation,
                if any to occur for calls exiting the network at this port.
                The tunnelling mode will screen the called address against
                the prefixes configured for this port.  If no match is
                found, no action is taken.  If a remote gateway address is
                configured for the matching prefix entry, the called party
                address will be tunnelled as a called subaddress and the
                configured remote gateway address will be inserted as the
                called party address.  If a local gateway address
                is configured for the matching prefix entry, the calling
                party address, if present, will be tunnelled as a calling
                subaddress and the configured local gateway address, if
                present, will be inserted as the calling party.
                The replacement mode will operate as does the tunnelling
                mode, except the original called and calling party
                addresses will be discarded. The two translation modes are
                used for address interworking between networks using E.164
                ATM Endsystem addresses and native ISDN E.164 addresses."
        ::= { svcConfigEntry 7 }

svcConfigIngressAddrXlateMode OBJECT-TYPE
        SYNTAX  INTEGER {
                    disabled (1),
                    tunnel(2),
                    translate-e164-native-to-nsap (4),
                    translate-e164-nsap-to-native (5)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "This object determines the type of address translation,
                if any to occur for calls entering the network at this
                port. The tunnelling mode reverses the actions of egress
                tunnelling.  If a called subaddress is present, it will
                replace the called party address, which will be
                discarded.  If a calling subaddress is present, it will
                replace the calling party address, which will be discarded.
                The two translation modes are used for address
                interworking between networks
                using E.164 ATM Endsystem addresses and native ISDN E.164
                addresses."
        ::= { svcConfigEntry 8 }

svcConfigCugEnable OBJECT-TYPE
        SYNTAX  INTEGER {
                disable (1),
                enable(2)
                 }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Enable/disable CUG processing at this port"
                DEFVAL { enable }
        ::= { svcConfigEntry 9 }

svcConfigSecScrIngressMode OBJECT-TYPE
        SYNTAX INTEGER {
                all-screens(1),    -- Security screening enabled.
                default-screen(2) -- Security screening disabled.
                 }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                 "This object determines whether the ingress screening is
                enabled at this port."
        DEFVAL {default-screen}
        ::= {svcConfigEntry 10}

svcConfigSecScrIngressDefaultScreen OBJECT-TYPE
        SYNTAX INTEGER {
                pass (1),  -- Pass all incoming calls.
                block (2)  -- Block all incoming calls.
                 }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                 "This object determines whether all incoming calls
                at this port should be passed or blocked. This setting is
                superseded by the security screens
                assigned at this port if the screen mode is set to all
                screens."
        DEFVAL {pass}
        ::= {svcConfigEntry 11}
```

```
svcConfigSecScrEgressMode OBJECT-TYPE
        SYNTAX INTEGER {
               all-screens (1),    -- Security screening enabled.
                default-screen (2) -- Security screening disabled.
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
               "This object determines whether the egress security
               screening is enabled at this port."
        DEFVAL {default-screen}
 ::= {svcConfigEntry 12}

svcConfigSecScrEgressDefaultScreen OBJECT-TYPE
        SYNTAX INTEGER {
                   pass  (1),  --Pass all outgoing calls.
                   block (2)   --Block all outgoing calls.
                   }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object determines whether all outgoing calls at
               this port should be passed or blocked. This setting is
               superseded by the security screens assign at this port if
               the screen mode is set to all screens."
        DEFVAL {pass}
        ::= {svcConfigEntry 13}

svcConfigSvcFailureLogReset OBJECT-TYPE
        SYNTAX INTEGER {
                        invalid (1), -- always returned for get
                        reset (2) -- reset the SVC failure log
                   }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object, when set to reset (2), will clear the SVC
               failure log in the switch."
        ::= {svcConfigEntry 14}

svcConfigSvcFailureTrapThreshold OBJECT-TYPE
        SYNTAX INTEGER (0..65535)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The threshold cross alarm value for SVC failure traps in
               the current 15 minute counting period.  When the internal
               SVC failure counter crosses this threshold, a trap will be
               generated.  The internal counter is reset every 15
               minutes."
        ::= {svcConfigEntry 15}

svcConfigNumSvcFailures OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The total number of SVC failures on this port, as
               defined by RELEASE, ADD PARTY REJECT or DROP PARTY  PDU's
               received or  transmitted with abnormal cause codes."
        ::= {svcConfigEntry 16}

svcConfigLoadBalanceEligibilityDuration OBJECT-TYPE
        SYNTAX INTEGER (0..65535)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The minimum duration for an SVC, in seconds, before it
               becomes eligible for load balancing.  The distinguished
               value, 0, indicates no load balancing for SVC's originating
               at this interface."
D       EFVAL { 3600 }
        ::= {svcConfigEntry 17}

svcConfigVpiVpciMappingType OBJECT-TYPE
        SYNTAX INTEGER {
                        equal (1),  -- vpi = vpci
                        positiveOffset (2), -- vpi = vpci + offset
                        negativeOffset (3), -- vpi = vpci - offset
                        table (4)  -- vpi = table(vpci)
                    }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "VPCI to VPI Mapping Type.  Given a VPCI, calculate
               the VPI"
               DEFVAL { equal }
        ::= {svcConfigEntry 18}

svcConfigVpiVpciMappingOffset OBJECT-TYPE
        SYNTAX INTEGER (0..4095)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "VPCI to VPI offset."
       DEFVAL { 0 }
        ::= {svcConfigEntry 19}
```

```
svcConfigProxyAdminStatus OBJECT-TYPE
        SYNTAX INTEGER {
                        disabled  (1),
                        proxy-agent (2),
                        proxy-client (3)
                    }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Proxy signaling administrative status
               disabled - no proxy capabilities
               proxy-agent - DCE UNI port only - lport acts as a
               network proxy agent.
               proxy-client - DCE UNI port only - signaling is
               performed by a peer PSA"
       DEFVAL { disabled }
        ::= {svcConfigEntry 20}

svcConfigProxyPSANodeId OBJECT-TYPE
        SYNTAX INTEGER (0..65535)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Node ID of the peer PSA - 0 equals local node."
       DEFVAL { 0 }
        ::= {svcConfigEntry 21}

svcConfigProxyPSAIfNum OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Interface number of the peer PSA - 0 equals local If
               num."
               DEFVAL { 0 }
        ::= {svcConfigEntry 22}
```

SVC ATM Configuration Table

```
svcAtmConfigTable OBJECT-TYPE
       SYNTAX  SEQUENCE OF SvcAtmConfigEntry
       ACCESS  not-accessible
       STATUS  mandatory
       DESCRIPTION
               "A table of ATM signalling paramaters associated with ATM
               logical ports.  The maximum number of entries is given by
               the value of ifNumber in MIB-II."
       ::= { svcmgt 2 }

svcAtmConfigEntry OBJECT-TYPE
       SYNTAX  SvcAtmConfigEntry
       ACCESS  not-accessible
       STATUS  mandatory
       DESCRIPTION
               "The SVC ATM configuration entry contains objects relevant
               to the configuration and monitoring of ATM signalling on
               an ATM logical port."
       INDEX   { svcAtmConfigIfIndex }
       ::= { svcAtmConfigTable 1 }
```

```
SvcAtmConfigEntry ::=
        SEQUENCE {
                svcAtmConfigIfIndex
                    Index,
                svcAtmConfigSigAdminStatus
                    INTEGER,
                svcAtmConfigSigOperStatus
                    INTEGER,
                svcAtmConfigQ93bMaxRestart
                    INTEGER,
                svcAtmConfigQ93bMaxStatEnq
                    INTEGER,
                svcAtmConfigQ93bT303
                    INTEGER,
                svcAtmConfigQ93bT308
                    INTEGER,
                svcAtmConfigQ93bT309
                    INTEGER,
                svcAtmConfigQ93bT310
                    INTEGER,
                svcAtmConfigQ93bT313
                    INTEGER,
                svcAtmConfigQ93bT316
                    INTEGER,
                svcAtmConfigQ93bT322
                    INTEGER,
                svcAtmConfigQ93bT398
                    INTEGER,
                svcAtmConfigQ93bT399
                    INTEGER,
                svcAtmConfigQ93bTotalConns
                    Counter,
                svcAtmConfigQ93bActiveConns
                    Counter,
                svcAtmConfigQ93bLastCauseTx
                    INTEGER,
                svcAtmConfigQ93bLastCauseRx
                    INTEGER,
                svcAtmConfigQ93bNumSetupPduTx
                    Counter,
                svcAtmConfigQ93bNumSetupPduRx
                    Counter,
                svcAtmConfigQ93bNumCallProcPduTx
                    Counter,
                svcAtmConfigQ93bNumCallProcPduRx
                    Counter,
                svcAtmConfigQ93bNumConnectPduTx
                    Counter,
                svcAtmConfigQ93bNumConnectPduRx
                    Counter,
                svcAtmConfigQ93bNumConnectAckPduTx
                    Counter,
                svcAtmConfigQ93bNumConnectAckPduRx
                    Counter,
                svcAtmConfigQ93bNumReleasePduTx
                    Counter,
                svcAtmConfigQ93bNumReleasePduRx
                    Counter,
                svcAtmConfigQ93bNumReleaseCmpltPduTx
                    Counter,
                svcAtmConfigQ93bNumReleaseCmpltPduRx
                    Counter,
                svcAtmConfigQ93bNumAddPtyPduTx
                    Counter,
                svcAtmConfigQ93bNumAddPtyPduRx
                    Counter,
                svcAtmConfigQ93bNumAddPtyAckPduTx
                    Counter,
                svcAtmConfigQ93bNumAddPtyAckPduRx
                    Counter,
                svcAtmConfigQ93bNumAddPtyRejPduTx
                    Counter,
                svcAtmConfigQ93bNumAddPtyRejPduRx
                    Counter,
                svcAtmConfigQ93bNumDropPtyPduTx
                    Counter,
                svcAtmConfigQ93bNumDropPtyPduRx
                    Counter,
                svcAtmConfigQ93bNumDropPtyAckPduTx
                    Counter,
                svcAtmConfigQ93bNumDropPtyAckPduRx
                    Counter,
                svcAtmConfigQ93bNumStatusEnqPduTx
                    Counter,
                svcAtmConfigQ93bNumStatusEnqPduRx
                    Counter,
                svcAtmConfigQ93bNumStatusPduTx
                    Counter,
                svcAtmConfigQ93bNumStatusPduRx
                    Counter,
                svcAtmConfigQ93bNumRestartPduTx
                    Counter,
                svcAtmConfigQ93bNumRestartPduRx
                    Counter,
                svcAtmConfigQ93bNumRestartAckPduTx
                    Counter,
                svcAtmConfigQ93bNumRestartAckPduRx
                    Counter,
                svcAtmConfigQSaalMaxCC
                    INTEGER,
                svcAtmConfigQSaalMaxPD
                    INTEGER,
                svcAtmConfigQSaalMaxStat
                    INTEGER,
                svcAtmConfigQSaalTPoll
                    INTEGER,
                svcAtmConfigQSaalTKeepalive
                    INTEGER,
```

```
svcAtmConfigQSaalTNoResponse
    INTEGER,
svcAtmConfigQSaalTCC
    INTEGER,
svcAtmConfigQSaalTIdle
    INTEGER,
svcAtmConfigQSaalNumDiscardTx
    Counter,
svcAtmConfigQSaalNumDiscardRx
    Counter,
svcAtmConfigQSaalNumErrorTx
    Counter,
svcAtmConfigQSaalNumErrorRx
    Counter,
svcAtmConfigQSaalNumBgnPduTx
    Counter,
svcAtmConfigQSaalNumBgnPduRx
    Counter,
svcAtmConfigQSaalNumBgakPduTx
    Counter,
svcAtmConfigQSaalNumBgakPduRx
    Counter,
svcAtmConfigQSaalNumBgrejPduTx
    Counter,
svcAtmConfigQSaalNumBgrejPduRx
    Counter,
svcAtmConfigQSaalNumEndPduTx
    Counter,
svcAtmConfigQSaalNumEndPduRx
    Counter,
svcAtmConfigQSaalNumEndakPduTx
    Counter,
svcAtmConfigQSaalNumEndakPduRx
    Counter,
svcAtmConfigQSaalNumRsPduTx
    Counter,
svcAtmConfigQSaalNumRsPduRx
    Counter,
svcAtmConfigQSaalNumRsakPduTx
    Counter,
svcAtmConfigQSaalNumRsakPduRx
    Counter,
svcAtmConfigQSaalNumErPduTx
    Counter,
svcAtmConfigQSaalNumErPduRx
    Counter,
svcAtmConfigQSaalNumErakPduTx
    Counter,
svcAtmConfigQSaalNumErakPduRx
    Counter,
svcAtmConfigQSaalNumSdPduTx
    Counter,
svcAtmConfigQSaalNumSdPduRx
    Counter,
svcAtmConfigQSaalNumPollPduTx
    Counter,
svcAtmConfigQSaalNumPollPduRx
    Counter,
svcAtmConfigQSaalNumStatPduTx
    Counter,
svcAtmConfigQSaalNumStatPduRx
    Counter,
svcAtmConfigQSaalNumUstatPduTx
    Counter,
svcAtmConfigQSaalNumUstatPduRx
    Counter,
svcAtmConfigQSaalNumUdPduTx
    Counter,
svcAtmConfigQSaalNumUdPduRx
    Counter,
svcAtmConfigQSaalNumMdPduTx
    Counter,
svcAtmConfigQSaalNumMdPduRx
    Counter,
svcAtmConfigQSaalNumOctetsTx
    Counter,
svcAtmConfigQSaalNumOctetsRx
    Counter
svcAtmConfigVpiStartVp
    INTEGER,
svcAtmConfigVpiStopVp
    INTEGER,
svcAtmConfigVpiStart
    INTEGER,
svcAtmConfigVpiStop
    INTEGER,
 svcAtmConfigVciStart
    INTEGER,
 svcAtmConfigVciStop
    INTEGER,
svcAtmConfigQSaalWindowSize
    INTEGER,
svcAtmConfigSvcCdvt
    INTEGER
}
}

svcAtmConfigIfIndex OBJECT-TYPE
       SYNTAX  Index
       ACCESS  read-only
       STATUS  mandatory
       DESCRIPTION
              "The ifIndex value of the  corresponding ifEntry."
       ::= { svcAtmConfigEntry 1 }
```

```
svcAtmConfigSigAdminStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                enabled (1),
                disabled (2)
        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The configured state of the ATM signalling function for
                this port."
        ::= { svcAtmConfigEntry 2 }

svcAtmConfigSigOperStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                down (1),
                connecting (2),
                up (3)
        }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The operational status of the signalling function on this
                port."
        ::= { svcAtmConfigEntry 3 }

svcAtmConfigQ93bMaxRestart OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The maximum number of unacknowledged restarts to send
                before declaring a signalling failure."
        ::= { svcAtmConfigEntry 4 }

svcAtmConfigQ93bMaxStatEnq OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The maximum number of unacknowledged status enquiries to
                send before issuing a restart."
        ::= { svcAtmConfigEntry 5 }

svcAtmConfigQ93bT303 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Protocol Timer T303, specified in milliseconds."
        ::= { svcAtmConfigEntry 6 }

svcAtmConfigQ93bT308 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Protocol Timer T308, specified in milliseconds."
        ::= { svcAtmConfigEntry 7 }

svcAtmConfigQ93bT309 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Protocol Timer T309, specified in milliseconds."
        ::= { svcAtmConfigEntry 8 }
svcAtmConfigQ93bT310 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Protocol Timer T310, specified in milliseconds."
        ::= { svcAtmConfigEntry 9 }

svcAtmConfigQ93bT313 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Protocol Timer T313, specified in milliseconds."
        ::= { svcAtmConfigEntry 10 }

svcAtmConfigQ93bT316 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Protocol Timer T316, specified in milliseconds."
        ::= { svcAtmConfigEntry 11 }

svcAtmConfigQ93bT322 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Protocol Timer T322, specified in milliseconds."
        ::= { svcAtmConfigEntry 12 }

svcAtmConfigQ93bT398 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Protocol Timer T398, specified in milliseconds."
        ::= { svcAtmConfigEntry 13 }
```

```
svcAtmConfigQ93bT399 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Protocol Timer T399, specified in milliseconds."
        ::= { svcAtmConfigEntry 14 }

svcAtmConfigQ93bTotalConns OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of SVC's established on this port."
        ::= { svcAtmConfigEntry 15 }

svcAtmConfigQ93bActiveConns OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of SVC's present on this port."
        ::= { svcAtmConfigEntry 16 }
```

```
svcAtmConfigQ93bLastCauseTx OBJECT-TYPE
        SYNTAX  INTEGER {
                unalloc-nmb     (1),        -- unallocated (unassigned)
                                               number
                no-route-transnet(2),       -- no route to transit network
                no-route-dest   (3),        -- no route to destination
                vcc-unacceptable-30(10),    -- UNI 3.0: VPCI/VCI unacceptable
                normal-call-clr-31(16),     -- UNI 3.1: normal call clearing
                user-busy       (17),       -- user busy
                no-user-response(18),       -- no user response
                call-reject     (21),       -- call rejected
                nmb-changed     (22),       -- number changed
                call-reject-clir(23),       -- user rejects all calls with
                                               CLIR
                dest-out-of-order(27),      -- destination out of order
                invalid-nmb-format(28),     -- invalid number format
                response-stat-enq(30),      -- response to STATUS ENQUIRY
                normal-unspecified(31),     -- normal unspecified
                req-vcc-unavailable(35),    -- requested VPCI/VCI unavailable
                vcc-fail-31     (36),       -- UNI 3.1: VPCI/VCI assignment
                                               failure
                rate-unavail-31 (37),       -- UNI 3.1: user cell rate
                                               unavailable
                network-out-of-order(38),   -- network out of order
                temp-fail       (41),       -- Temporary failure
                access-info-discard(43),    -- access info discarded
                no-vcc-available(45),       -- no VPCI/VCI unavailable
                resources-unavailable(47),  -- resources unavailable,
                                               unspecified
                qos-unavailable (49),       -- Quality of Service unavailable
                rate-unavailable-30(51),    -- UNI 3.0: user cell rate
                                               unavailable
                b-cap-not-authorized(57),   -- bearer capability not
                                               authorized
                b-cap-unavailable(58),      -- bearer capability not
                                               available
                service-unavailable(63),    -- Service or option unavailable
```

```
                b-cap-not-implemented(65),-- bearer capability not
                                          implemented
                combination-unsupported(73)-- unsupported comb. of
                                          traffic parameters
                aal-params-unsupp-31(78),  -- UNI 3.1: AAL paramteres cannot
                                          be supported
                invalid-call-reference(81),-- invalid call reference
                no-channel(82),           -- identified channel does not
                                          exist
                dest-incompatible(88),    -- incompatible destination
                invalid-endpoint-ref(89), -- invalid endpoint reference
                invalid-transit-net(91),  -- invalid transit network
                                          selection
                too-many-add-pty-req(92), -- too many add party requests
                aal-params-unsupp-30(93), -- UNI 3.0:AAL paramteres cannot
                                          be supported
                info-element-missing(96), -- mandatory info element is
                                          missing
                msg-type-not-imp(97),     -- message type not implemented
                info-element-not-imp(99), -- info element not implemented
                invalid-info-element(100),-- invalid info element
                message-not-compatible(101),-- msg type not compatible
                                            with call st
                timer-recovery (102),     -- recovery on timer expiry
                invalid-message-len(104), -- incorrect message length
                protocol-error (111),     -- protocol error, unspecified
                optional-element-error(127)-- opt info el content error
                                          (non-std)
        }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The last transmitted cause code for this port."
        ::= { svcAtmConfigEntry 17 }

svcAtmConfigQ93bLastCauseRx OBJECT-TYPE
        SYNTAX  INTEGER {
                unalloc-nmb    (1),       -- unallocated (unassigned
                )                         number
                no-route-transnet(2),     -- no route to transit network
                no-route-dest  (3),       -- no route to destination
                vcc-unacceptable-30(10),  -- UNI 3.0: VPCI/VCI unacceptable
                normal-call-clr-31(16),   -- UNI 3.1: normal call clearing
                user-busy      (17),      -- user busy
                no-user-response(18),     -- no user response
                call-reject    (21),      -- call rejected
                nmb-changed    (22),      -- number changed
                call-reject-clir(23),     -- user rejects all calls with
                                          CLIR
                dest-out-of-order(27),    -- destination out of order
                invalid-nmb-format(28),   -- invalid number format
                response-stat-enq(30),    -- response to STATUS ENQUIRY
                normal-unspecified(31),   -- normal unspecified
                req-vcc-unavailable(35),  -- requested VPCI/VCI unavailable
                vcc-fail-31    (36),      -- UNI 3.1: VPCI/VCI assignment
                                          failure
                rate-unavail-31 (37),     -- UNI 3.1: user cell rate
                                          unavailable
                network-out-of-order(38), -- network out of order
                temp-fail      (41),      -- Temporary failure
                access-info-discard(43),  -- access info discarded
                no-vcc-available(45),     -- no VPCI/VCI unavailable
                resources-unavailable(47),-- resources unavailable,
                                          unspecified
                qos-unavailable (49),     -- Quality of Service unavailable
                rate-unavailable-30(51),  -- UNI 3.0: user cell rate
                                          unavailable
                b-cap-not-authorized(57), -- bearer capability not
                                          authorized
                b-cap-unavailable(58),    -- bearer capability not
                                          available
                service-unavailable(63),  -- Service or option unavailable
                b-cap-not-implemented(65),-- bearer capability not
                                          implemented
                combination-unsupported (73),-- unsupported comb. of
                                             traffic parameters
                aal-params-unsupp-31(78), -- UNI 3.1: AAL paramteres cannot
                                          be supported
                invalid-call-reference(81),-- invalid call reference
                no-channel     (82),      -- identified channel does not
                                          exist
                dest-incompatible(88),    -- incompatible destination
                invalid-endpoint-ref(89), -- invalid endpoint reference
                invalid-transit-net(91),  -- invalid transit network
                                          selection
                too-many-add-pty-req(92), -- too many add party requests
                aal-params-unsupp-30(93), -- UNI 3.0:AAL paramteres cannot
                                          be supported
                info-element-missing(96), -- mandatory info element is
                                          missing
                msg-type-not-imp(97),     -- message type not implemented
                info-element-not-imp(99), -- info element not implemented
                invalid-info-element(100),-- invalid info element
                message-not-compatible(101),-- msg type not compatible
                                             with call st
                timer-recovery (102),     -- recovery on timer expiry
                invalid-message-len(104), -- incorrect message length
                protocol-error (111),     -- protocol error, unspecified
                optional-element-error(127)-- opt info el content error
                                          (non-std)
        }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The last received cause code for this port."
        ::= { svcAtmConfigEntry 18 }
```

```
svcAtmConfigQ93bNumSetupPduTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of setup PDU's transmitted on this port."
        ::= { svcAtmConfigEntry 19 }

svcAtmConfigQ93bNumSetupPduRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of setup PDU's received on this port."
        ::= { svcAtmConfigEntry 20 }

svcAtmConfigQ93bNumCallProcPduTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of call proceeding PDU's transmitted on this
                port."
        ::= { svcAtmConfigEntry 21 }

svcAtmConfigQ93bNumCallProcPduRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of call proceeding PDU's received on this
                port."
        ::= { svcAtmConfigEntry 22 }

svcAtmConfigQ93bNumConnectPduTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of connect PDU's transmitted on this port."
        ::= { svcAtmConfigEntry 23 }

svcAtmConfigQ93bNumConnectPduRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of connect PDU's received on this port."
        ::= { svcAtmConfigEntry 24 }

svcAtmConfigQ93bNumConnectAckPduTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of connect ack PDU's transmitted on this port."
        ::= { svcAtmConfigEntry 25 }

svcAtmConfigQ93bNumConnectAckPduRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of connect ack PDU's received on this port."
        ::= { svcAtmConfigEntry 26 }

svcAtmConfigQ93bNumReleasePduTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of release PDU's transmitted on this port."
        ::= { svcAtmConfigEntry 27 }

svcAtmConfigQ93bNumReleasePduRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of release PDU's received on this port."
        ::= { svcAtmConfigEntry 28 }

svcAtmConfigQ93bNumReleaseCmpltPduTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of release complete PDU's transmitted on this
                port."
        ::= { svcAtmConfigEntry 29 }

svcAtmConfigQ93bNumReleaseCmpltPduRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of release complete PDU's received on this
                port."
        ::= { svcAtmConfigEntry 30 }
```

```
svcAtmConfigQ93bNumAddPtyPduTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of add party PDU's transmitted on this port."
        ::= { svcAtmConfigEntry 31 }

svcAtmConfigQ93bNumAddPtyPduRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of add party PDU's received on this port."
        ::= { svcAtmConfigEntry 32 }

svcAtmConfigQ93bNumAddPtyAckPduTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of add party acknowledge PDU's transmitted on
                this port."
        ::= { svcAtmConfigEntry 33 }

svcAtmConfigQ93bNumAddPtyAckPduRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of add party acknowledge PDU's received on
                this port."
        ::= { svcAtmConfigEntry 34 }

svcAtmConfigQ93bNumAddPtyRejPduTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of add party reject PDU's transmitted on this
                port."
        ::= { svcAtmConfigEntry 35 }

svcAtmConfigQ93bNumAddPtyRejPduRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of add party reject PDU's received on this
                port."
        ::= { svcAtmConfigEntry 36 }

svcAtmConfigQ93bNumDropPtyPduTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of drop party PDU's transmitted on this port."
        ::= { svcAtmConfigEntry 37 }

svcAtmConfigQ93bNumDropPtyPduRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of drop party PDU's received on this port."
        ::= { svcAtmConfigEntry 38 }

svcAtmConfigQ93bNumDropPtyAckPduTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of drop party acknowledge PDU's transmitted on
                this port."
        ::= { svcAtmConfigEntry 39 }

svcAtmConfigQ93bNumDropPtyAckPduRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of drop party acknowledge PDU's received on
                this port."
        ::= { svcAtmConfigEntry 40 }

svcAtmConfigQ93bNumStatusEnqPduTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of status enquiry PDU's transmitted on this
                port."
        ::= { svcAtmConfigEntry 41 }

svcAtmConfigQ93bNumStatusEnqPduRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of status enquiry PDU's received on this port."
        ::= { svcAtmConfigEntry 42 }
```

```
svcAtmConfigQ93bNumStatusPduTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of status PDU's transmitted on this port."
        ::= { svcAtmConfigEntry 43 }

svcAtmConfigQ93bNumStatusPduRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of status PDU's received on this port."
        ::= { svcAtmConfigEntry 44 }

svcAtmConfigQ93bNumRestartPduTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of restart PDU's transmitted on this port."
        ::= { svcAtmConfigEntry 45 }

svcAtmConfigQ93bNumRestartPduRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of restart PDU's received on this port."
        ::= { svcAtmConfigEntry 46 }

svcAtmConfigQ93bNumRestartAckPduTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of restart acknowledge PDU's transmitted on
                this port."
        ::= { svcAtmConfigEntry 47 }

svcAtmConfigQ93bNumRestartAckPduRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of restart acknowledge PDU's received on this
                port."
        ::= { svcAtmConfigEntry 48 }

svcAtmConfigQSaalMaxCC OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The maximum number of unacknowledged transmitted control
                PDU's before declaring a loss of connection."
        ::= { svcAtmConfigEntry 49 }

svcAtmConfigQSaalMaxPD OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The maximum number of PDU's transmitted before a POLL PDU
                is transmitted."
        ::= { svcAtmConfigEntry 50 }

svcAtmConfigQSaalMaxStat OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The maximum number of list elements in a STAT PDU."
        ::= { svcAtmConfigEntry 51 }

svcAtmConfigQSaalTPoll OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The protocol timer corresponding to the polling function,
                 specified in milliseconds."
        ::= { svcAtmConfigEntry 52 }

svcAtmConfigQSaalTKeepalive OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The protocol timer corresponding to the keepalive
                function, specified in milliseconds."
        ::= { svcAtmConfigEntry 53 }

svcAtmConfigQSaalTNoResponse OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The protocol timer corresponding to the response timeout
                function, specified in milliseconds."
        ::= { svcAtmConfigEntry 54 }
```

```
svcAtmConfigQSaalTCC OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The protocol timer corresponding to the transmission of
                 control PDU's, specified in milliseconds."
        ::= { svcAtmConfigEntry 55 }

svcAtmConfigQSaalTIdle OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The protocol timer corresponding to the idle function for
                 UNI 3.1, only, specified in milliseconds."
        ::= { svcAtmConfigEntry 56 }

svcAtmConfigQSaalNumDiscardTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of QSaal transmit discards on this port."
        ::= { svcAtmConfigEntry 57 }

svcAtmConfigQSaalNumDiscardRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of QSaal receive discards on this port."
        ::= { svcAtmConfigEntry 58 }

svcAtmConfigQSaalNumErrorTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of QSaal transmit errors on this port."
        ::= { svcAtmConfigEntry 59 }

svcAtmConfigQSaalNumErrorRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of QSaal receive errors on this port."
        ::= { svcAtmConfigEntry 60 }

svcAtmConfigQSaalNumBgnPduTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of begin PDU's transmitted on this port."
        ::= { svcAtmConfigEntry 61 }

svcAtmConfigQSaalNumBgnPduRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of begin PDU's received on this port."
        ::= { svcAtmConfigEntry 62 }

svcAtmConfigQSaalNumBgakPduTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of begin acknowledge PDU's transmitted on this
                port."
        ::= { svcAtmConfigEntry 63 }

svcAtmConfigQSaalNumBgakPduRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of begin acknowledge PDU's received on this
                port."
        ::= { svcAtmConfigEntry 64 }

svcAtmConfigQSaalNumBgrejPduTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of begin reject PDU's transmitted on this
                port."
        ::= { svcAtmConfigEntry 65 }

svcAtmConfigQSaalNumBgrejPduRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of begin reject PDU's received on this port."
        ::= { svcAtmConfigEntry 66 }
```

```
svcAtmConfigQSaalNumEndPduTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of end PDU's transmitted on this port."
        ::= { svcAtmConfigEntry 67 }

svcAtmConfigQSaalNumEndPduRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of end PDU's received on this port."
        ::= { svcAtmConfigEntry 68 }

svcAtmConfigQSaalNumEndakPduTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of end acknowledge PDU's transmitted on this
                port."
        ::= { svcAtmConfigEntry 69 }

svcAtmConfigQSaalNumEndakPduRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of end acknowldge PDU's received on this port."
        ::= { svcAtmConfigEntry 70 }

svcAtmConfigQSaalNumRsPduTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of resynchronization PDU's transmitted on this
                port."
        ::= { svcAtmConfigEntry 71 }

svcAtmConfigQSaalNumRsPduRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of resynchronization PDU's received on this
                port."
        ::= { svcAtmConfigEntry 72 }

svcAtmConfigQSaalNumRsakPduTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of resynchronization acknowledge PDU's
                transmitted on this port."
        ::= { svcAtmConfigEntry 73 }

svcAtmConfigQSaalNumRsakPduRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of resynchronization acknowldge PDU's received
                on this port."
        ::= { svcAtmConfigEntry 74 }

svcAtmConfigQSaalNumErPduTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of error recovery PDU's transmitted on this
                port."
        ::= { svcAtmConfigEntry 75 }

svcAtmConfigQSaalNumErPduRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of errror recovery PDU's received on this
                port."
        ::= { svcAtmConfigEntry 76 }

svcAtmConfigQSaalNumErakPduTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of error recovery acknowledge PDU's
                transmitted on this port."
        ::= { svcAtmConfigEntry 77 }

svcAtmConfigQSaalNumErakPduRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of errror recovery acknowledge PDU's received
                on this port."
        ::= { svcAtmConfigEntry 78 }
```

```
svcAtmConfigQSaalNumSdPduTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of sequenced data PDU's transmitted on this
                port."
        ::= { svcAtmConfigEntry 79 }

svcAtmConfigQSaalNumSdPduRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of sequenced data PDU's received on this port."
        ::= { svcAtmConfigEntry 80 }

svcAtmConfigQSaalNumPollPduTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of poll PDU's transmitted on this port."
        ::= { svcAtmConfigEntry 81 }

svcAtmConfigQSaalNumPollPduRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of poll PDU's received on this port."
        ::= { svcAtmConfigEntry 82 }

svcAtmConfigQSaalNumStatPduTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of status PDU's transmitted on this port."
        ::= { svcAtmConfigEntry 83 }

svcAtmConfigQSaalNumStatPduRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of status PDU's received on this port."
        ::= { svcAtmConfigEntry 84 }

svcAtmConfigQSaalNumUstatPduTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of unsolicited status PDU's transmitted on
                this port."
        ::= { svcAtmConfigEntry 85 }

svcAtmConfigQSaalNumUstatPduRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of unsolicited status PDU's received on this
                port."
        ::= { svcAtmConfigEntry 86 }

svcAtmConfigQSaalNumUdPduTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of unnumbered user data PDU's transmitted on
                this port."
        ::= { svcAtmConfigEntry 87 }

svcAtmConfigQSaalNumUdPduRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of unnumbered user data PDU's received on this
                port."
        ::= { svcAtmConfigEntry 88 }

svcAtmConfigQSaalNumMdPduTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of unnumbered management data PDU's
                transmitted on this port."
        ::= { svcAtmConfigEntry 89 }

svcAtmConfigQSaalNumMdPduRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of unnumbered management data PDU's received
                on this port."
        ::= { svcAtmConfigEntry 90 }
```

```
svcAtmConfigQSaalNumOctetsTx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of octets sent on the signalling channel."
        ::= { svcAtmConfigEntry 91 }

svcAtmConfigQSaalNumOctetsRx OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of octets received from the signalling
                channel."
        ::= { svcAtmConfigEntry 92 }

svcAtmConfigVpiStartVp OBJECT-TYPE
        SYNTAX INTEGER (0..4095)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Starting VPI value for SVP range.  Must be less than or
                equal to svcAtmConfigVpiStopVp.  This object is used in
                conjunction with the svcAtmConfigVpiStopVp object.  Values
                of 0 for both objects invalidates the range checks."
        ::= { svcAtmConfigEntry 93 }

svcAtmConfigVpiStopVp OBJECT-TYPE
        SYNTAX INTEGER (0..4095)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Ending VPI value for SVP range.  Must be greater than
                or equal to svcAtmConfigVpiStartVp. This object is used
                in conjunction with the svcAtmConfigVpiStartVp object.
                Values of 0 for both objects invalidates the range checks."
        ::= { svcAtmConfigEntry 94 }

svcAtmConfigVpiStart OBJECT-TYPE
        SYNTAX INTEGER (0..4095)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Starting VPI value for SVC range.  Must be less than or
                equal to svcAtmConfigVpiStop. This object is used in
                conjunction with the svcAtmConfigVpiStop,
                svcAtmConfigVciStart and
                svcAtmConfigVciStop objects.  Values of 0 for all objects
                invalidates the range checks"
        ::= { svcAtmConfigEntry 95 }

svcAtmConfigVpiStop OBJECT-TYPE
        SYNTAX INTEGER (0..4095)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Ending VPI value for SVC range.  Must be greater than
                or equal to svcAtmConfigVpiStart. This object is used in
                conjunction with the svcAtmConfigVpiStart,
                svcAtmConfigVciStart and
                svcAtmConfigVciStop objects.  Values of 0 for all objects
                invalidates the range checks."
        ::= { svcAtmConfigEntry 96 }

svcAtmConfigVciStart OBJECT-TYPE
        SYNTAX INTEGER (32..65535)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Starting VCI value for SVC range.  Must be less than or
                 equal to svcAtmConfigVciStop. This object is used in
                conjunction with the svcAtmConfigVpiStart,
                svcAtmConfigVpiStop and
                svcAtmConfigVciStop objects.  Values of 0 for all objects
                invalidates the range checks."
        ::= { svcAtmConfigEntry 97}

svcAtmConfigVciStop OBJECT-TYPE
        SYNTAX INTEGER (32..65535)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Ending VCI value for SVC range.  Must be greater than of
                equal to svcAtmConfigVciStart. This object is used in
                conjunction with the svcAtmConfigVpiStart,
                svcAtmConfigVpiStop and
                svcAtmConfigVciStart objects.  Values of 0 for all objects
                invalidates the range checks."
        ::= { svcAtmConfigEntry 98 }

svcAtmConfigQSaalWindowSize OBJECT-TYPE
        SYNTAX INTEGER (1..4095)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The window size for Q.SAAL."
        DEFVAL { 64 }
        ::= { svcAtmConfigEntry 99 }
```

```
svcAtmConfigSvcCdvt OBJECT-TYPE
        SYNTAX INTEGER (1..16777215)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The default cell delay variation tolerance, in
                microseconds, to be used by the UPC at this port for
                originating or terminating SVC's.  Higher values will
                result in a less restrictive UPC."
        DEFVAL { 200 }
        ::= { svcAtmConfigEntry 100 }

--  The Screen Table for Security Screening.
--

screenTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF ScreenTableEntry
        ACCESS not-accessible
        STATUS  mandatory
        DESCRIPTION
  "A table of configuration parameters associated with a screen."
 ::= { svcmgt 3}

screenTableEntry OBJECT-TYPE
        SYNTAX ScreenTableEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "The screen table entry contains objects relevant to the
                screen parameters"
        INDEX {secScrLport, secScrScrID}
        ::= { screenTable 1 }

ScreenTableEntry ::=
        SEQUENCE {
                secScrLport
                    Index,
                secScrScrID
                    INTEGER,
                secScrCgPtyType
                    INTEGER,
                secScrCgPty
                    OCTET STRING,
                secScrCdPtyType
                    INTEGER,
                secScrCdPty
                    OCTET STRING,
                secScrCgPtySubAddrType
                    INTEGER,
                secScrCgPtySubAddr
                    OCTET STRING,
                secScrCdPtySubAddrType
                    INTEGER,
                secScrCdPtySubAddr
                    OCTET STRING,
                secScrDirection
                    INTEGER,
                secScrType
                    INTEGER,
                secScrAdminState
                    INTEGER
                }

secScrLport OBJECT-TYPE
        SYNTAX Index
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The ifIndex value of the corresponding
                ifEntry."
        ::= { screenTableEntry 1 }

secScrScrID OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS mandatory
        DESCRIPTION
                "The screen index to access the screens applied at a port.
                The maximum number of screens applied to a port is limited
                to 16."
        ::= { screenTableEntry 2 }

secScrCgPtyType OBJECT-TYPE
        SYNTAX INTEGER {
                disable (1),  -- Cg Pty screening disabled.
                atm-endsystem (2), -- AESA addressing format.
                e164 (4)  -- E.164 addressing format.
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object specifies the addressing format. Supported
                addressing formats are AESA and E.164. The screening can
                be disabled by setting it to disable."
        DEFVAL {disable}
        ::= { screenTableEntry 3 }

secScrCgPty OBJECT-TYPE
        SYNTAX OCTET STRING
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object specifies the calling party address to be
                used by the security screen for matching."
        ::= { screenTableEntry 4 }
```

```
secScrCdPtyType OBJECT-TYPE
        SYNTAX  INTEGER {
                disable (1),             -- Cd Pty screening disabled.
                atm-endsystem (2),       -- AESA addressing format.
                e164 (4)                 -- E.164 addressing format.
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "This object specifies the addressing format. Supported
                addressing formats are AESA and E.164. The screening can
                be disabled by setting it to disable."
        DEFVAL {disable}
        ::= { screenTableEntry 5 }

secScrCdPty     OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "This object specifies the called party address to be
                used by the security screen for matching."
        ::= { screenTableEntry 6 }

secScrCgPtySubAddrType OBJECT-TYPE
        SYNTAX  INTEGER {
        disable (1),  -- Disabled.
        atm-endsystem (2) -- AESA addressing format.
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "This object enables/disables the calling party subaddress
                screening. Supported addressing format is AESA."
        DEFVAL {disable}
        ::= { screenTableEntry 7 }

secScrCgPtySubAddr     OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "This object specifies the calling party subaddress to be
                used by the security screen for matching."
        ::= { screenTableEntry 8 }
```

```
secScrCdPtySubAddrType OBJECT-TYPE
        SYNTAX  INTEGER {
                disable (1),             -- Disabled.
                atm-endsystem (2)        -- AESA addressing format.
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "This object enables/disables the called party subaddress
                screening. Supported addressing format is AESA."
        DEFVAL {disable}
        ::= { screenTableEntry 9 }

secScrCdPtySubAddr     OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "This object specifies the called party subaddress to be
                used by the security screen for matching."
        ::= { screenTableEntry 10 }

secScrDirection OBJECT-TYPE
        SYNTAX INTEGER {
                ingress (1), -- incoming call screen.
                egress (2) -- outgoing call screen.
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object specifies the direction of the screen."
        DEFVAL {ingress}
        ::= {screenTableEntry 11}


secScrType OBJECT-TYPE
        SYNTAX INTEGER {
                pass (1),  -- screen is a pass screen.
                block (2) -- screen is a block screen.
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object specifies the action to be taken on the call
                if it meets the matching criteria."
        DEFVAL {pass}
        ::= { screenTableEntry 12 }
```

```
secScrAdminState OBJECT-TYPE
        SYNTAX INTEGER {
                invalid   (0), -- Screen not valid.
                active   (1), -- Screen activated.
                inactive  (2)   -- Screen inactivated.
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object determines if the screen information is valid
                to use. This should be the last object to be set to
                active, while all the screen pertaining information
                is updated. And it should be set to inactive when the
                screen is to be ignored for decision making purpose. It
                should be set to 'delete' once the screen is deleted."
        DEFVAL {invalid}
 ::= {screenTableEntry 13}
```

SVC - Virtual Path Channel Identifier Table

```
svcVpciTable OBJECT-TYPE
        SYNTAX SEQUENCE OF SvcVpciEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "A table of VPCIs associated with ports on this node."
        ::= { svcmgt 4 }

svcVpciEntry OBJECT-TYPE
        SYNTAX SvcVpciEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "The VPCI entry contains objects relevant to VPCIs
               associated with this port."
        INDEX { svcVpciIfIndex, svcVpciVpci }
        ::= { svcVpciTable 1 }
```

```
SvcVpciEntry ::=
        SEQUENCE {
               svcVpciIfIndex
                  Index,
                svcVpciVpci
                  INTEGER,
                svcVpciVpi
                  INTEGER,
                svcVpciTargetNodeId
                  INTEGER,
               svcVpciTargetInterface
                  INTEGER,
               svcVpciVciAssignmentPolicy
                  INTEGER,
               svcVpciRowStatus
                  INTEGER
                }

svcVpciIfIndex OBJECT-TYPE
        SYNTAX Index
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The interface value of the corresponding MIB-II ifEntry."
        ::= { svcVpciEntry 1 }

svcVpciVpci OBJECT-TYPE
        SYNTAX INTEGER (0..65535)
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "VPCI is an abstract 16 bit value which is used to
               identify the connection resources associated with an
               interface."
        ::= { svcVpciEntry 2 }

svcVpciVpi OBJECT-TYPE
        SYNTAX INTEGER (0..4095)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Actual value specified here is used in the ATM Cell
               Header. At an NNI interface VPI is 12-Bits and at the
               UNI the VPI is 8 Bits. The value of VPI which can
               be selected is limited to the VPI range specified at the
               lport.
               Range of VCIs which may be allocated for this VPI is
               specified in svcVciStart & svcVciStop."
        DEFVAL { 0 }
        ::= { svcVpciEntry 3 }
```

```
svcVpciTargetNodeId OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "If a signalling entity signals on behalf of other
               interfaces then those target node ID must be
               explicitly identified.  A node ID of 0
                means local node."
        DEFVAL { 0 }
        ::= { svcVpciEntry 4 }

svcVpciTargetInterface OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "If a signalling entity signals on behalf of other
               interfaces then those target interface must be
               explicitly identified.  A target interface of 0
                 means local interface."
        DEFVAL { 0 }
        ::= { svcVpciEntry 5 }

svcVpciVciAssignmentPolicy OBJECT-TYPE
        SYNTAX INTEGER {
                   all (1), -- assign all VCIs for this VPCI;
                   none (2), -- assign none of the VCIs for this VPCI;
                   even (3), -- assign all even VCIs from the VCI range;
                   odd  (4), -- assign all odd VCIs from the VCI range;
                   unspecified (5) -- higher level protocol will decide
                                      the policy;
                   }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object defines the assignment rule which must be
               observed by the signalling entity such as UNI or
               BISUP - Signalling."
        DEFVAL { all }
        ::= { svcVpciEntry 6 }

svcVpciRowStatus OBJECT-TYPE
        SYNTAX INTEGER {
                   unknown (1), -- Row status has not been set yet;
                   active (2), -- Row is active;
                   inactive (3), -- Row has been created, but is inactive;
                   delete (4) -- Row is to be removed
                   }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Status of row in VPCI table."
        DEFVAL { unknown }
        ::= { svcVpciEntry 7 }
```

The SVC CUG Group

```
--  The tables that are relevant to managing Closed User Groups in a
Cascade network.
```

CUG Table

```
closedUserGroupTable OBJECT-TYPE
        SYNTAX SEQUENCE OF ClosedUserGroupEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "A list of all Closed User Groups active an a
                particular node"
        ::= { svccug 1 }

closedUserGroupEntry OBJECT-TYPE
        SYNTAX ClosedUserGroupEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "The CUG entry contains all information for a
                particular CUG"
        INDEX { cugIdentifier }
 ::= { closedUserGroupTable 1 }

ClosedUserGroupEntry ::=
        SEQUENCE {
                cugIdentifier
                    INTEGER,
                cugAdminStatus
                    INTEGER,
                cugNumMembers
                    INTEGER
                }
```

```
cugIdentifier OBJECT-TYPE
        SYNTAX INTEGER (1..65535)
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Global CUG Identifier"
        ::= { closedUserGroupEntry 1 }

cugAdminStatus OBJECT-TYPE
        SYNTAX INTEGER {
                configured (1),
                invalid (2)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The administrative status of this entry"
        DEFVAL { configured }
        ::= { closedUserGroupEntry 2 }

cugNumMembers OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Number of members in CUG"
        ::= { closedUserGroupEntry 3 }
```

CUG Member Table

```
cugMemberTable OBJECT-TYPE
        SYNTAX SEQUENCE OF CugMemberEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "A list of all CUG Members"
        ::= { svccug 2 }

cugMemberEntry OBJECT-TYPE
        SYNTAX CugMemberEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "The member entry contains all information for a
                particular member"
        INDEX { cugMemberIndex }
        ::= { cugMemberTable 1 }

CugMemberEntry ::=
        SEQUENCE {
                cugMemberIndex
                    INTEGER,
                cugMemberRule
                    OCTET STRING,
                cugMemberAdminStatus
                    INTEGER,
                cugMemberNmbPlan
                    INTEGER,
                cugMemberOutgoingAccess
                    INTEGER,
                cugMemberIncomingAccess
                    INTEGER,
                cugMemberNumCugs
                    INTEGER
                }

cugMemberIndex OBJECT-TYPE
        SYNTAX INTEGER (1..65535)
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Global Member Index"
        ::= { cugMemberEntry 1 }

cugMemberRule OBJECT-TYPE
        SYNTAX OCTET STRING
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Member Rule.  Once the administrative status has
                changed to configured, then this object CANNOT be changed
                to a different value.  If the rule must change after
                the row has become configured, then the whole row must
                be deleted and re-created with the new value."
        ::= { cugMemberEntry 2 }

cugMemberAdminStatus OBJECT-TYPE
        SYNTAX INTEGER {
                configured (1),
                invalid (2)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The administrative status of this entry"
        DEFVAL { configured }
        ::= { cugMemberEntry 3 }
```

```
cugMemberNmbPlan OBJECT-TYPE
        SYNTAX INTEGER {
                e164 (1),
                atm-endsystem (2),
                unknown (3)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The numbering plan corresponding to this entry. Once
                the administrative status has changed to configured,
                then this object CANNOT be changed to a different
                value.  If the numbering plan must change after
                the row has become configured, then the whole row must
                be deleted and re-created with the new value."
        DEFVAL { unknown }
        ::= { cugMemberEntry 4 }

cugMemberOutgoingAccess OBJECT-TYPE
        SYNTAX INTEGER {
                no (1),
                yes (2)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Outgoing access attribute"
        DEFVAL { no }
        ::= { cugMemberEntry 5 }

cugMemberIncomingAccess OBJECT-TYPE
        SYNTAX INTEGER {
                no (1),
                yes (2)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Incoming access attribute"
        DEFVAL { no }
        ::= { cugMemberEntry 6 }

cugMemberNumCugs OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Number of CUGs member to which member belongs"
        ::= { cugMemberEntry 7 }
```

CUG Member to CUG association Table

```
cugMemberCugListTable OBJECT-TYPE
        SYNTAX SEQUENCE OF CugMemberCugListEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "A list of all CUG Members"
        ::= { svccug 3 }

cugMemberCugListEntry OBJECT-TYPE
        SYNTAX CugMemberCugListEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "The member entry contains all information for a
                particular member"
        INDEX { mcMemberIndex, mcCugIdentifier }
        ::= { cugMemberCugListTable 1 }

CugMemberCugListEntry ::=
        SEQUENCE {
                mcMemberIndex
                    INTEGER,
                mcCugIdentifier
                    INTEGER,
                mcAdminStatus
                    INTEGER,
                mcIncomingCallsBarred
                    INTEGER,
                mcOutgoingCallsBarred
                    INTEGER,
                mcPreferentialCug
                    INTEGER
                }

mcMemberIndex OBJECT-TYPE
        SYNTAX INTEGER (0..65535)
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Member Rule Index"
        ::= { cugMemberCugListEntry 1 }
```

```
mcCugIdentifier OBJECT-TYPE
        SYNTAX INTEGER (0..65535)
        ACCESS read-only
        TATUS mandatory
        DESCRIPTION
                "Global CUG Identifier"
        ::= { cugMemberCugListEntry 2 }

mcAdminStatus OBJECT-TYPE
        SYNTAX INTEGER {
                configured (1),
                invalid (2)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The administrative status of this entry"
        DEFVAL { configured }
        ::= { cugMemberCugListEntry 3 }

mcIncomingCallsBarred OBJECT-TYPE
        SYNTAX INTEGER {
                no (1),
                yes (2)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Incoming calls barred attribute"
        DEFVAL { no }
        ::= { cugMemberCugListEntry 4 }

mcOutgoingCallsBarred OBJECT-TYPE
        SYNTAX INTEGER {
                no (1),
                yes (2)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Outgoing calls barred attribute"
        DEFVAL { no }
        ::= { cugMemberCugListEntry 5 }

mcPreferentialCug OBJECT-TYPE
        SYNTAX INTEGER {
                no (1),
                yes (2)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Preferential CUG attribute.  Will be used with
                signaling support"
        DEFVAL { no }
        ::= { cugMemberCugListEntry 6 }
```

The Software Group

```
--  The variables that describe the software running on a particular card.

swTable  OBJECT-TYPE
        SYNTAX  SEQUENCE OF SwEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A list of swEntry's. The number of entries is given by
                the value of swCardNumber"
        ::= { software 1 }

swEntry  OBJECT-TYPE
        SYNTAX  SwEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "The software entry contains objects describing the
                software cards."
        INDEX       { swLogicalSlotId, swRedundState }
        ::= { swTable 1 }
```

```
SwEntry  ::=
        SEQUENCE {
                swLogicalSlotId
                    INTEGER,
                swRedundState
                    INTEGER,
                swRevision
                    DisplayString,
                swBuildID
                    DisplayString,
                swBuildDate
                    DisplayString,
                swBuildDescription
                    DisplayString,
                swCopyrightNotice
                    DisplayString,
                swCapabilityMask
                    INTEGER,
                swFeatureMask
                    INTEGER,
                swPatchMask
                    INTEGER,
                swBuildUserId
                    DisplayString,
                swBuildView
                    DisplayString,
                swBuildConfigSpec
                    DisplayString
        }

swLogicalSlotId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The logical slot number of this card. This is used as an
                index for the swTable. In redundant configurations, this
                can be the physical slot number of either redundant card.
                In non redundant configurations, this is the same as
                cardPhysicalSlotId."
        ::= { swEntry 1}

swRedundState OBJECT-TYPE
        SYNTAX  INTEGER {
                    active (1),
                    standby (2)
                    }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The current redundancy state of this card."
        ::= { swEntry 2 }

swRevision OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The software revision number in the form
                (major.minor.maint.patch)
                where major == the major release number
                      minor == the minor release number
                      maint == a maintenance release based on major.minor
                      patch == a patch release based on major.minor.maint"
        ::= { swEntry 3 }

swBuildID OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "An identifier which uniquely identifies the software
                image within the scope of the release.  The build ID is
                generated automatically through the build process and
                assigned to the software image during the final load
                generation."
        ::= { swEntry 4 }

swBuildDate OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The date on which the software image was generated."
        ::= { swEntry 5 }

swBuildDescription OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A general description of the release."
        ::= { swEntry 6 }

swCopyrightNotice OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Legalese describing the Copyrights covering this software
                image."
        ::= { swEntry 7 }
```

```
swCapabilityMask OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A 32-bit integer which bitmaps optional or additional
                features supported by this software release.  For use by
                the NMS in determining capabilities of the software.  The
                significance of this bitmap varies by release.For Cascade
                internal use."
        ::= { swEntry 8 }

swFeatureMask OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A 32-bit integer which bitmaps optional or additional
                features supported by this software release.  For use by
                the NMS in determining capabilities of the software.  The
                significance of this bitmap varies by release.For Cascade
                internal use."
        ::= { swEntry 9 }

swPatchMask OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A 32-bit integer which bitmaps incremental patches which
                have been applied to the software release.  The
                significance of this bitmap varies by release.For Cascade
                internal use."
        ::= { swEntry 10 }

swBuildUserId OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The user ID of the person responsible for generating the
                software image. For Cascade internal use."
        ::= { swEntry 11 }

swBuildView OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Source control information for image generation.  For
                Cascade internal use."
        ::= { swEntry 12 }

swBuildConfigSpec OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Source control information for image generation.  For
                Cascade internal use."
        ::= { swEntry 13 }
```

SVC DTE User Part Table

```
svcDteUserPartTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF SvcDteUserPartEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A table of partial SVC address associated with ports on
                this node, relevant only to DTE ports for use in ILMI
                address registration."
        ::= { svcaddress 4 }

svcDteUserPartEntry OBJECT-TYPE
        SYNTAX  SvcDteUserPartEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "The user part entry contains objects relevant to partial
                SVC addresses associated with DTE ports on this node."
        INDEX { svcDteUserPartIfIndex, svcDteUserPartUserPart }
        ::= { svcDteUserPartTable 1 }

SvcDteUserPartEntry ::=
        SEQUENCE {
                svcDteUserPartIfIndex
                    Index,
                svcDteUserPartUserPart
                    OCTET STRING,
                svcDteUserPartNumDigits
                    INTEGER,
                svcDteUserPartNmbPlan
                    INTEGER,
                svcDteUserPartAdminStatus
                    INTEGER
        }

svcDteUserPartIfIndex OBJECT-TYPE
        SYNTAX  Index
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The primary index for this entry is the interface value
                of the corresponding MIB-II ifEntry."
        ::= { svcDteUserPartEntry 1 }
```

```
svcDteUserPartUserPart OBJECT-TYPE
        SYNTAX OCTET STRING (SIZE(8..20))
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The secondary index for this entry is a left-justified, 0-
                padded, partial SVC address.  E.164 entries require an 8
                octet string while SVC End System entries require a 20
                octet string."
        ::= { svcDteUserPartEntry 2 }

svcDteUserPartNumDigits OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The number of significant BCD digits, if numbering plan is
                E.164, or the number of significant octets, if numbering
                plan is SVC End System, 0 for this entry."
        ::= { svcDteUserPartEntry 3 }

svcDteUserPartNmbPlan OBJECT-TYPE
        SYNTAX  INTEGER {
                e164 (1),
                atmEndSystem (2),
                unknown (4)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The numbering plan used for this SVC address UserPart
                entry."
        ::= { svcDteUserPartEntry 4 }

svcDteUserPartAdminStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                configured (1),-- this entry has been configured by NMS
                invalid (2)    -- this entry shall be deleted
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The status of the SVC address UserPart entry."
        ::= { svcDteUserPartEntry 5 }
```

The SVC Remote Prefix Table

```
svcRemotePrefixTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF SvcRemotePrefixEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A table of remote ATM Endsystem prefixes associated with
                this port for purposes of static routing an SVC from one
                private network to another via a public transit network
                supporting only ISDN addressing."
        ::= { svcaddress 5 }

svcRemotePrefixEntry OBJECT-TYPE
        SYNTAX  SvcRemotePrefixEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "The prefix entry contains objects relevant to remote
                prefixes associated with the port."
        INDEX { svcRemotePrefixIfIndex, svcRemotePrefixPrefix }
        ::= { svcRemotePrefixTable 1 }

SvcRemotePrefixEntry ::=
        SEQUENCE {
                svcRemotePrefixIfIndex
                    Index,
                svcRemotePrefixPrefix
                    OCTET STRING,
                svcRemotePrefixNumDigits
                    INTEGER,
                svcRemotePrefixNmbPlan
                    INTEGER,
                svcRemotePrefixLocalGatewayAddress
                    OCTET STRING,
                svcRemotePrefixLocalGatewayNmbPlan
                    INTEGER,
                svcRemotePrefixRemoteGatewayAddress
                    OCTET STRING,
                svcRemotePrefixRemoteGatewayNmbPlan
                    INTEGER,
                svcRemotePrefixAdminStatus
                    INTEGER
                }

svcRemotePrefixIfIndex OBJECT-TYPE
        SYNTAX  Index
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The primary index for this entry is the interface value
                of the corresponding MIB-II ifEntry."
        ::= { svcRemotePrefixEntry 1 }
```

```
svcRemotePrefixPrefix OBJECT-TYPE
        SYNTAX  OCTET STRING (SIZE(20))
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The secondary index for this entry is a left-justified, 0-
                padded, ATM Endsystem SVC remote prefix."
        ::= { svcRemotePrefixEntry 2 }

svcRemotePrefixNumDigits OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The number of significant octets for this entry's
                svcRemotePrefixPrefix."
        ::= { svcRemotePrefixEntry 3 }

svcRemotePrefixNmbPlan OBJECT-TYPE
        SYNTAX  INTEGER {
                e164 (1),
                atmEndSystem (2),
                unknown(3)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The numbering plan used for this remote prefix entry."
        ::= { svcRemotePrefixEntry 4 }

svcRemotePrefixLocalGatewayAddress OBJECT-TYPE
        SYNTAX  OCTET STRING (SIZE(8..20))
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The ISDN address to use at this port for this remote
                prefix as a calling address.  If set to 0, an attempt will
                be made to find an address matching the
                svcRemotePrefixLocalGatewayNmbPlan from the
                svcAddrTable.  If none is found, calling address will be
                omitted for calls to this entry's remote prefix."
        ::= { svcRemotePrefixEntry 5 }

svcRemotePrefixLocalGatewayNmbPlan OBJECT-TYPE
        SYNTAX  INTEGER {
                e164 (1),
                atmEndSystem (2),
                unknown (4)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The numbering plan associated with the local gateway
                address for this entry.  Also, the numbering plan to use
                when searching the svcAddrTable if
                svcRemotePrefixLocalGatewayAddress is set to 0."
        ::= { svcRemotePrefixEntry 6 }

svcRemotePrefixRemoteGatewayAddress OBJECT-TYPE
        SYNTAX  OCTET STRING (SIZE(8..20))
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The ISDN address to use at this port for this remote
                prefix as a called address."
        ::= { svcRemotePrefixEntry 7 }

svcRemotePrefixRemoteGatewayNmbPlan OBJECT-TYPE
        SYNTAX  INTEGER {
                e164 (1),
                atmEndSystem (2),
                unknown (4)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The numbering plan associated with the local gateway
                address for this entry."
        ::= { svcRemotePrefixEntry 8 }

svcRemotePrefixAdminStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                configured (1), -- this entry has been configured by NMS
                invalid (2)    -- this entry shall be deleted
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The status of this entry."
        ::= { svcRemotePrefixEntry 9 }
```

SVC Failed Call Table

```
svcFailedCallTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF SvcFailedCallEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A table of objects describing the N most recent SVC
                failures per port, in FIFO order."
        ::= { svcmgt 2 }

svcFailedCallEntry OBJECT-TYPE
        SYNTAX  SvcFailedCallEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "The failed call entry contains objects relevant to
                abnormal SVC call failures."
        INDEX { svcFailedCallIfIndex, svcFailedCallIndex }
        ::= { svcFailedCallTable 1 }

SvcFailedCallEntry ::=
        SEQUENCE {
                svcFailedCallIfIndex
                    Index,
                svcFailedCallIndex
                    INTEGER,
                svcFailedCallPduType
                    INTEGER,
                svcFailedCallCause
                    INTEGER,
                svcFailedCallLocation
                    INTEGER,
                svcFailedCallDiagnostic
                    OCTET STRING,
                svcFailedCallCreationTime
                    TimeTicks,
                svcFailedCallTerminationTime
                    TimeTicks,
                svcFailedCallFailureNodeId
                    INTEGER,
                svcFailedCallFailureIfIndex
                    Index,
                svcFailedCallCallingParty
                    OCTET STRING,
                svcFailedCallCalledParty
                    OCTET STRING,
                svcFailedCallAtmTfdType
                    INTEGER,
                svcFailedCallAtmTfdParam1
                    INTEGER,
                svcFailedCallAtmTfdParam2
                    INTEGER,
                svcFailedCallAtmTfdParam3
                    INTEGER,
                svcFailedCallAtmQosClass
                    INTEGER,
                svcFailedCallAtmRTfdType
                    INTEGER,
                svcFailedCallAtmRTfdParam1
                    INTEGER,
                svcFailedCallAtmRTfdParam2
                    INTEGER,
                svcFailedCallAtmRTfdParam3
                    INTEGER,
                svcFailedCallAtmRQoSClass
                    INTEGER,
                svcFailedCallAdminStatus
                    INTEGER
                }

svcFailedCallIfIndex OBJECT-TYPE
        SYNTAX  Index
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The interface ID at the ingress port of the corresponding
                failed call."
        ::= { svcFailedCallEntry 1 }

svcFailedCallIndex OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The instance number for this failed call."
        ::= { svcFailedCallEntry 2 }

svcFailedCallPduType OBJECT-TYPE
        SYNTAX  INTEGER {
                release(1),
                addPartyReject(2),
                dropParty(3)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The Pdu Type containing the cause IE, if an ATM SVC."
        ::= { svcFailedCallEntry 3 }
```

```
svcFailedCallCause OBJECT-TYPE
        SYNTAX  INTEGER {
                unallocNmb (1),     -- unallocated (unassigned) number
                noRoutTransnet (2), -- no route to transit network
                noRoutDst (3),      -- no route to destination
                vccUnacpt (10),     -- UNI 3.0: VPCI/VCI unacceptable
                normCallClr (16),   -- UNI 3.1: normal call clearing
                usrBusy (17),       -- user busy
                nousrRsp (18),      -- no user response
                callRej (21),       -- call rejected
                nmbChng (22),       -- number changed
                callRejClir (23),   -- user rejects all calls with CLIR
                dstOutOrd (27),     -- destination out of order
                invNmbForm (28),    -- invalid number format
                rspStatEnq (30),    -- response to STATUS ENQUIRY
                normUnspec (31),    -- normal unspecified
                reqVccUnavail (35), -- requested VPCI/VCI unavailable
                vccFail (36),       -- UNI 3.1: VPCI/VCI assignment failure
                rateUnavail1 (37),  -- UNI 3.1: user cell rate unavailable
                netOutOrd (38),     -- network out of order
                tmpFail (41),       -- Temporary failure
                accInfoDisc (43),   -- access info discarded
                noVccAvail (45),    -- no VPCI/VCI unavailable
                resUnavail (47),    -- resources unavailable, unspecified
                qosUnavail (49),    -- Quality of Service unavailable
                rateUnavail (51),   -- UNI 3.0: user cell rate unavailable
                bCapNotAuth (57),   -- bearer capability not authorized
                bCapUnavail (58),   -- bearer capability not available
                srvUnavail (63),    -- Service or option unavailable
                bCapNotImpl (65),   -- bearer capability not implemented
                combUnsupp (73),    -- unsupported comb. of traffic
                                    -- parameters
                aalParmUnsupp1 (78),-- UNI 3.1: AAL paramters cannot be
                                    -- supported
                invCallRef (81),    -- invalid call reference
                chanNotExst (82),   -- identified channel does not exist
                dstNotComp (88),    -- incompatible destination
                invEndptRef (89),   -- invalid endpoint reference
                invTransNet (91),   -- invalid transit network selection
                manyAddptyReq (92), -- too many add party requests
                aalParmUnsupp (93), -- UNI 3.0:AAL paramteres cannot be
                                    -- supported
                infoElMssg (96),    -- mandatory info element is missing
                msgTypNotImpl (97), -- message type not implemented
                infoElNotImpl (99), -- info element not implemented
                invInfoEl (100),    -- invalid info element
                msgNotComp (101),   -- msg type not compatible with call st
                tmrRcvry (102),     -- recovery on timer expiry
                invMsgLen (104),    -- incorrect message length
                protErr (111),      -- protocol error, unspecified
                optElErr (127)      -- opt info el content error (non-std)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The failure cause contained in the cause IE."
        ::= { svcFailedCallEntry 4 }

svcFailedCallLocation OBJECT-TYPE
        SYNTAX  INTEGER {
                privNetServLocal (1),  -- private network serving local user
                pubNetServLocal (2),   -- public network serving local user
                transitNet (3),        -- transit network
                pubNetServRemote (4),  -- public network serving remote user
                privNetServRemote (5), -- private network serving remote
                                       -- user
                intlNet (6),           -- international network
                netBeyondInterwkPt (7),-- network beyond interworking point
                user (8)               -- user
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The failure location contained in the cause IE."
        ::= { svcFailedCallEntry 5 }

svcFailedCallDiagnostic OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The diagnostic information contained in the cause IE."
        ::= { svcFailedCallEntry 6 }

svcFailedCallCreationTime OBJECT-TYPE
        SYNTAX  TimeTicks
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The sysUpTime when this call was initiated."
        ::= { svcFailedCallEntry 7 }

svcFailedCallTerminationTime OBJECT-TYPE
        SYNTAX  TimeTicks
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The sysUpTime when this call was terminated."
        ::= { svcFailedCallEntry 8 }

svcFailedCallFailureNodeId OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The node which caused the SVC failure."
        ::= { svcFailedCallEntry 9 }
```

```
svcFailedCallFailureIfIndex OBJECT-TYPE
        SYNTAX  Index
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The port on the failure node which causes the SVC
                failure."
        ::= { svcFailedCallEntry 10 }

svcFailedCallCallingParty OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The calling party number for this failed call."
        ::= { svcFailedCallEntry 11 }

svcFailedCallCalledParty OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The called party number for this failed call."
        ::= { svcFailedCallEntry 12 }

svcFailedCallAtmTfdType OBJECT-TYPE
        SYNTAX  INTEGER {
                pcr-0-01 (1),
                pcr-0-01-tag (2),
                pcr-01-scr-0-mbs-0 (3),
                pcr-01-scr-0-mbs-0-tag (4),
                pcr-01 (5),
                pcr-01-scr-01-mbs-01 (6),
                pcr-01-bestEffort (7)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The combination of forward traffic parameters for this
                failed call."
        ::= { svcFailedCallEntry 13 }

svcFailedCallAtmTfdParam1 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Forward traffic descriptor parameter 1 for this failed
                call."
        ::= { svcFailedCallEntry 14 }

svcFailedCallAtmTfdParam2 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Forward traffic descriptor parameter 2 for this failed
                call."
        ::= { svcFailedCallEntry 15 }

svcFailedCallAtmTfdParam3 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Forward traffic descriptor parameter 3 for this failed
                call."
        ::= { svcFailedCallEntry 16 }

svcFailedCallAtmQosClass OBJECT-TYPE
        SYNTAX  INTEGER {
                cbr (1),        -- constant bit rate
                vbr1 (2),       -- variable bit rate, real-time
                vbr2 (3),       -- variable bit rate, non-real-time
                vbr3 (4)        -- unspecified/available bit rate
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Forward QOS class for this failed call."
        ::= { svcFailedCallEntry 17 }

svcFailedCallAtmRTfdType OBJECT-TYPE
        SYNTAX  INTEGER {
                pcr-0-01 (1),
                pcr-0-01-tag (2),
                pcr-01-scr-0-mbs-0 (3),
                pcr-01-scr-0-mbs-0-tag (4),
                pcr-01 (5),
                pcr-01-scr-01-mbs-01 (6),
                pcr-01-bestEffort (7)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The combination of reverse traffic parameters for this
                failed call."
        ::= { svcFailedCallEntry 18 }
```

```
svcFailedCallAtmRTfdParam1 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Reverse traffic descriptor parameter 1 for this failed
                call."
        ::= { svcFailedCallEntry 19 }

svcFailedCallAtmRTfdParam2 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Reverse traffic descriptor parameter 2 for this failed
                call."
        ::= { svcFailedCallEntry 20 }

svcFailedCallAtmRTfdParam3 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Reverse traffic descriptor parameter 3 for this failed
                call."
        ::= { svcFailedCallEntry 21 }

svcFailedCallAtmRQoSClass OBJECT-TYPE
        SYNTAX  INTEGER {
                cbr (1),        -- constant bit rate
                vbr1 (2),       -- variable bit rate, real-time
                vbr2 (3),       -- variable bit rate, non-real-time
                vbr3 (4)        -- unspecified/available bit rate
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Reverse QOS class for this failed call."
        ::= { svcFailedCallEntry 22 }

svcFailedCallAdminStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                valid (1),      -- entry is valid
                invalid (2)     -- entry is invalid
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "All entries are initially invalid.  Due to the FIFO
                nature of this table, once N entries have been created,
                all entries will always be valid."
        ::= { svcFailedCallEntry 23 }
```

Cascade Performance Monitoring MIBs

```
--      These branches consist of MIB data required for performance
        monitoring that are
--      not supported in the standard DS1, DS3, or SONET MIBs.  These
        MIBs consist of
--      separate DS1, DS3, and SONET branches, all off if the cascadepm
        branch.
--      DS1 Performance Monitoring Delta MIB

--      This MIB consists of delta objects that supplement the standard
        DS1 MIB to
--      support DS1 Performance Monitoring.  These supplements include:
--      o  ANSI T1.231 Support - increased set of counters, and
        thresholds.
--      Consists of the following tables
--      The Configuration Table
--      Current Table
--      Interval Table
--      Total Table
--      Current Threshold Table (15 minutes)
--      Day Threshold Table (24 hours)
```

DS1 PM Delta Configuration Table

```
ds1pmConfigTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF Ds1pmConfigEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "The DS1 PM Configuration table."
        ::= { ds1pm 1 }

ds1pmConfigEntry OBJECT-TYPE
        SYNTAX  Ds1pmConfigEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "An entry in the DS1 PM Configuration table."
        INDEX   { ds1pmConfigInterfaceIndex }
        ::= { ds1pmConfigTable 1 }
```

```
Ds1pmConfigEntry ::=
        SEQUENCE {
            ds1pmConfigInterfaceIndex
                 INTEGER,
            ds1pmConfigValidTotals
                 INTEGER,
            ds1pmConfigResetCurrent
                 INTEGER,
            ds1pmConfigResetInterval
                 INTEGER,
            ds1pmConfigResetTotal
                 INTEGER,
            ds1pmConfigThresholdCrossingEnable
                 INTEGER,
            ds1pmConfigAlarmSoakTime
                 INTEGER,
            ds1pmConfigAlarmClearTime
                 INTEGER
               }

ds1pmConfigInterfaceIndex OBJECT-TYPE
        SYNTAX  INTEGER (1..'7fffffff'h)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This object is the identifier of an Inter-
                face on a managed device.  If there is an ifEn-
                try that is directly associated with  this  and
                only  this interface,  it should have the
                same value as ifIndex.   Otherwise,  the  value
                exceeds  ifNumber,  and  is a unique identifier
                following this rule: inside  interfaces  (e.g.,
                equipment  side)  with even numbers and outside
                interfaces  (e.g.,  network   side)   with   odd
                numbers."
        ::= { ds1pmConfigEntry 1 }

ds1pmConfigValidTotals OBJECT-TYPE
        SYNTAX  INTEGER (0..2)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of  previous  total intervals  for  which
                valid data was collected.  The value will be 2
                unless the interface was brought on-line within
                the last 2 days, in which case the value will
                be the number of complete 24 hour  intervals
                the since interface has been online."
        ::= { ds1pmConfigEntry 2 }

ds1pmConfigResetCurrent OBJECT-TYPE
        SYNTAX  INTEGER {
                   noRreset (1),
                   resetCurrent (2)
                   }

        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Reset the current accumulation registers to 0."
        ::= { ds1pmConfigEntry 3 }

ds1pmConfigResetInterval OBJECT-TYPE
        SYNTAX  INTEGER (0..31)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
               "Reset the interval accumulation registers to 0.  The
               interval number  of the interval to be cleared is
               given.  A selection of 0 clears the entire interval table."
        ::= { ds1pmConfigEntry 4 }

ds1pmConfigResetTotal OBJECT-TYPE
        SYNTAX  INTEGER (0..3)
        ACCESS read-write
        STATUS  mandatory
        DESCRIPTION
                " Reset the total accumulation registers to 0.  The
               interval number of the interval to be cleared is given.  A
               selection of 0 clears the entire total table."
        ::= { ds1pmConfigEntry 5 }

ds1pmConfigThresholdCrossingEnable OBJECT-TYPE
        SYNTAX  INTEGER {
                   disabled (1),
                   enabled (2)
                        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Enable or disabled the detection and emission of
               threshold crossing alarms."
        ::= { ds1pmConfigEntry 6 }

ds1pmConfigAlarmSoakTime OBJECT-TYPE
        SYNTAX  INTEGER (0..65535)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This is the soak time for configurable alarms. An alarm
                 of this type must persist for this period before it is
                 declared.   The time is specified in millseconds units."
        ::= { ds1pmConfigEntry 7 }
```

```
ds1pmConfigAlarmClearTime OBJECT-TYPE
        SYNTAX  INTEGER (0..65535)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This is the soak time for configurable alarms. An alarm
               of this type must remain clear for this period before it
               is declared to be cleared.   The time is specified in
               millseconds units."
        ::= { ds1pmConfigEntry 8 }
```

The DS1 PM Delta Current Table

```
-- The table contains various statistics being
-- collected for the current 15 minute interval.

ds1pmCurrentTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF Ds1pmCurrentEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "The DS1 PM Current table."
        ::= { ds1pm 2 }

ds1pmCurrentEntry OBJECT-TYPE
        SYNTAX  Ds1pmCurrentEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "An entry in the DS1 PM Current table."
        INDEX   { ds1pmCurrentInterfaceIndex }
        ::= { ds1pmCurrentTable 1 }

Ds1pmCurrentEntry ::=
        SEQUENCE {
             ds1pmCurrentInterfaceIndex
                 INTEGER,
             ds1pmCurrentInvalid
                 INTEGER,
             ds1pmCurrentSESL
                 Gauge,
             ds1pmCurrentLOSSL
                 Gauge,
             ds1pmCurrentESAP
                 Gauge,
             ds1pmCurrentAISSP
                 Gauge,
             ds1pmCurrentFCP
                 Gauge,
             ds1pmCurrentESAPFE
                 Gauge,
             ds1pmCurrentFCPFE
                 Gauge
        }

ds1pmCurrentInterfaceIndex OBJECT-TYPE
        SYNTAX  INTEGER (1..'7fffffff'h)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The index value which uniquely identifies  the
                interface  to which this entry is applica-
                ble.  The interface identified by a  particular
                value  of  this  index is the same interface as
                identified by the same value as a ansi231LineIndex
                object instance."
        ::= { ds1pmCurrentEntry 1 }

ds1pmCurrentInvalid OBJECT-TYPE
        SYNTAX  INTEGER {
               data_invalid (1),
               data_valid (2)
                        }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current data invalid indicator."
        ::= { ds1pmCurrentEntry 2 }

ds1pmCurrentSESL OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current line severely errored seconds count."
        ::= { ds1pmCurrentEntry 3 }

ds1pmCurrentLOSSL OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current line loss of signal seconds count."
        ::= { ds1pmCurrentEntry 4 }

ds1pmCurrentESAP OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current path errored seconds type A count."
        ::= { ds1pmCurrentEntry 5 }
```

```
ds1pmCurrentAISSP OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current  path AIS seconds count."
        ::= { ds1pmCurrentEntry 6}

ds1pmCurrentFCP OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current  path failure count."
        ::= { ds1pmCurrentEntry 7 }

ds1pmCurrentESAPFE OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current far end path errored seconds type A count."
        ::= { ds1pmCurrentEntry 8}

ds1pmCurrentFCPFE OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current far end path failure count."
        ::= { ds1pmCurrentEntry 9 }
```

The DS1 PM Delta Interval Table

```
-- This table consists of entries that store history data for the last
-- 96 15 minute intervals

ds1pmIntervalTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF Ds1pmIntervalEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "The DS1 PM Interval table."
        ::= { ds1pm 3 }

ds1pmIntervalEntry OBJECT-TYPE
        SYNTAX   Ds1pmIntervalEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "An entry in the DS1 PM Interval table."
        INDEX   {  ds1pmIntervalInterfaceIndex, ds1pmIntervalNumber }
        ::= {  ds1pmIntervalTable 1 }

Ds1pmIntervalEntry ::=
        SEQUENCE {
             ds1pmIntervalInterfaceIndex
                 INTEGER,
             ds1pmIntervalNumber
                 INTEGER,
             ds1pmIntervalSESL
                 Gauge,
             ds1pmIntervalLOSSL
                 Gauge,
             ds1pmIntervalESAP
                 Gauge,
             ds1pmIntervalAISSP
                 Gauge,
             ds1pmIntervalFCP
                 Gauge,
             ds1pmIntervalESAPFE
                 Gauge,
             ds1pmIntervalFCPFE
                 Gauge
               }

ds1pmIntervalInterfaceIndex OBJECT-TYPE
        SYNTAX  INTEGER (1..'7fffffff'h)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The index value which uniquely identifies  the
                interface  to which this entry is applica-
                ble.  The interface identified by a  particular
                value  of  this  index is the same interface as
                identified by the same value as a ds1pmLineIndex
                object instance."
        ::= {  ds1pmIntervalEntry 1 }

ds1pmIntervalNumber OBJECT-TYPE
        SYNTAX  INTEGER (1..96)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "."
        ::= {  ds1pmIntervalEntry 2 }

ds1pmIntervalSESL OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Interval line severely errored seconds count."
        ::= {  ds1pmIntervalEntry 3 }
```

```
ds1pmIntervalLOSSL OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Interval loss of signal seconds count."
        ::= {ds1pmIntervalEntry 4 }

ds1pmIntervalESAP OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Interval path errored seconds type A count."
        ::= {  ds1pmIntervalEntry 5 }

ds1pmIntervalAISSP OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Interval path AIS seconds count."
        ::= {  ds1pmIntervalEntry 6}

ds1pmIntervalFCP OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Interval path failure count."
        ::= {  ds1pmIntervalEntry 7 }

ds1pmIntervalESAPFE OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Interval far end path errored seconds type A count."
        ::= {  ds1pmIntervalEntry 8}

ds1pmIntervalFCPFE OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Interval far end path failure count."
        ::= {  ds1pmIntervalEntry 9 }
```

The DS1 PM Delta Total Table

```
-- This table contains 24 hour history registers for the last three days,
current
-- previous, and recent

ds1pmTotalTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF Ds1pmTotalEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "The DS1 PM total table."
        ::= { ds1pm 4 }

ds1pmTotalEntry OBJECT-TYPE
        SYNTAX  Ds1pmTotalEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "An entry in the DS1 PM total table."
        INDEX   { ds1pmTotalInterfaceIndex, ds1pmTotalIntervalNumber }
        ::= { ds1pmTotalTable 1 }

Ds1pmTotalEntry ::=
        SEQUENCE {
            ds1pmTotalInterfaceIndex
                 INTEGER,
            ds1pmTotalIntervalNumber
                 INTEGER,
            ds1pmTotalStatus
                 INTEGER,
            ds1pmTotalCVL
                 Gauge,
            ds1pmTotalESL
                 Gauge,
            ds1pmTotalSESL
                 Gauge,
            ds1pmTotalLOSSL
                 Gauge,
            ds1pmTotalCVP
                 Gauge,
            ds1pmTotalESP
                 Gauge,
            ds1pmTotalESAP
                 Gauge,
            ds1pmTotalESBP
                 Gauge,
            ds1pmTotalSESP
                 Gauge,
            ds1pmTotalSASP
                 Gauge,
            ds1pmTotalAISSP
                 Gauge,
            ds1pmTotalCSSP
```

```
            Gauge,
        ds1pmTotalUASP
            Gauge,
        ds1pmTotalFCP
            Gauge,
        ds1pmTotalESLFE
            Gauge,
        ds1pmTotalCVPFE
            Gauge,
        ds1pmTotalESPFE
            Gauge,
        ds1pmTotalESAPFE
            Gauge,
        ds1pmTotalESBPFE
            Gauge,
        ds1pmTotalSESPFE
            Gauge,
        ds1pmTotalSEFSPFE
            Gauge,
        ds1pmTotalCSSPFE
            Gauge,
        ds1pmTotalUASPFE
            Gauge,
        ds1pmTotalFCPFE
            Gauge
          }

ds1pmTotalInterfaceIndex OBJECT-TYPE
        SYNTAX  INTEGER (1..'7fffffff'h)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The index value which uniquely identifies  the
                interface  to which this entry is applica-
                ble.  The interface identified by a  particular
                value  of  this  index is the same interface as
                identified by the same value as a ds1pmLineIndex
                object instance."
        ::= {   ds1pmTotalEntry 1 }

ds1pmTotalIntervalNumber OBJECT-TYPE
        SYNTAX  INTEGER (1..2)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "."
        ::= {   ds1pmTotalEntry 2 }

ds1pmTotalStatus OBJECT-TYPE
        SYNTAX  INTEGER {
               dataInvalid (1),
               dataValid (2)
               }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Identifies the validity of the total table data ."
        ::= {   ds1pmTotalEntry 3 }

ds1pmTotalCVL OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                ""
        ::= {   ds1pmTotalEntry 4 }

ds1pmTotalESL OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                ""
        ::= {   ds1pmTotalEntry 5 }

ds1pmTotalSESL OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                ""
        ::= {   ds1pmTotalEntry 6 }

ds1pmTotalLOSSL OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total line loss of signal seconds count."
        ::= {ds1pmTotalEntry 7 }

ds1pmTotalCVP OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                ""
        ::= {   ds1pmTotalEntry 8 }

ds1pmTotalESP OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                ""
        ::= {   ds1pmTotalEntry 9 }
```

```
ds1pmTotalESAP OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total path errored seconds type A count."
        ::= {   ds1pmTotalEntry 10 }

ds1pmTotalESBP OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                ""
        ::= {   ds1pmTotalEntry 11 }

ds1pmTotalSESP OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                ""
        ::= {   ds1pmTotalEntry 12 }

ds1pmTotalSASP OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                ""
        ::= {   ds1pmTotalEntry 13 }

ds1pmTotalAISSP OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total path AIS seconds count."
        ::= {   ds1pmTotalEntry 14 }

ds1pmTotalCSSP OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                ""
        ::= {   ds1pmTotalEntry 15 }

ds1pmTotalUASP OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                ""
        ::= {   ds1pmTotalEntry 16 }

ds1pmTotalFCP OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total path failure count."
        ::= {   ds1pmTotalEntry 17 }

ds1pmTotalESLFE OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                ""
        ::= {   ds1pmTotalEntry 18 }

ds1pmTotalCVPFE OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                ""
        ::= {   ds1pmTotalEntry 19 }

ds1pmTotalESPFE OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                ""
        ::= {   ds1pmTotalEntry 20 }

ds1pmTotalESAPFE OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total far end path errored seconds type A count."
        ::= {   ds1pmTotalEntry 21 }
```

```
ds1pmTotalESBPFE OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                ""
        ::= {   ds1pmTotalEntry 22 }

ds1pmTotalSESPFE OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                ""
        ::= {   ds1pmTotalEntry 23 }

ds1pmTotalSEFSPFE OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                ""
        ::= {   ds1pmTotalEntry 24 }

ds1pmTotalCSSPFE OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                ""
        ::= {   ds1pmTotalEntry 25

ds1pmTotalUASPFE OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                ""
        ::= {   ds1pmTotalEntry 26 }

ds1pmTotalFCPFE OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total far end path failure count."
        ::= {   ds1pmTotalEntry 27 }
```

The DS1 PM Current Threshold Table

```
-- This table contains 15 minute (current) threshold values used
-- in performance parameter thresholding defined by ANSI T1.231

ds1pmCurrentThresholdTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF Ds1pmCurrentThresholdEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "The DS1 PM CurrentThreshold table."
        ::= { ds1pm 5 }

ds1pmCurrentThresholdEntry OBJECT-TYPE
        SYNTAX  Ds1pmCurrentThresholdEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "An entry in the DS1 PM Threshold table."
        INDEX   { ds1pmCurrentThresholdInterfaceIndex }
        ::= { ds1pmCurrentThresholdTable 1 }

Ds1pmCurrentThresholdEntry ::=
        SEQUENCE {
               ds1pmCurrentThresholdInterfaceIndex
                 INTEGER,
               ds1pmThreshESLCurrent
                 INTEGER,
               ds1pmThreshCVPCurrent
                 INTEGER,
               ds1pmThreshESPCurrent
                 INTEGER,
               ds1pmThreshSESPCurrent
                 INTEGER,
               ds1pmThreshSASPCurrent
                 INTEGER,
               ds1pmThreshCSSPCurrent
                 INTEGER,
               ds1pmThreshUASPCurrent
                 INTEGER
               }
```

```
ds1pmCurrentThresholdInterfaceIndex OBJECT-TYPE
        SYNTAX  INTEGER (1..'7fffffff'h)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The index value which uniquely identifies  the
                interface  to which this entry is applica-
                ble.  The interface identified by a  particular
                value  of  this  index is the same interface as
                identified by the same value as a ds1pmLineIndex
                object instance."
        ::= {   ds1pmCurrentThresholdEntry 1 }

ds1pmThreshESLCurrent OBJECT-TYPE
        SYNTAX  INTEGER (1..900)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Line errored seconds 15 Minute (current) threshold."
        ::= { ds1pmCurrentThresholdEntry 2 }

ds1pmThreshCVPCurrent OBJECT-TYPE
        SYNTAX  INTEGER (1..16383)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Path code violations 15 Minute (current) threshold."
        ::= { ds1pmCurrentThresholdEntry 3 }

ds1pmThreshESPCurrent OBJECT-TYPE
        SYNTAX  INTEGER (1..900)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Path errored seconds 15 Minute (current) threshold."
        ::= { ds1pmCurrentThresholdEntry 4 }

ds1pmThreshSESPCurrent OBJECT-TYPE
        SYNTAX  INTEGER (1..63)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Path severely errored seconds 15 Minute (current)
               threshold."
        ::= { ds1pmCurrentThresholdEntry 5 }

ds1pmThreshSASPCurrent OBJECT-TYPE
        SYNTAX  INTEGER (1..63)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Path SEF/AIS seconds 15 Minute (current) threshold."
        ::= { ds1pmCurrentThresholdEntry 6 }

ds1pmThreshCSSPCurrent OBJECT-TYPE
        SYNTAX  INTEGER (1..63)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Path controlled slip seconds 15 Minute (current)
               threshold."
        ::= { ds1pmCurrentThresholdEntry 7 }

ds1pmThreshUASPCurrent OBJECT-TYPE
        SYNTAX  INTEGER (1..63)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Path unavailable seconds 15 Minute (current) threshold."
        ::= { ds1pmCurrentThresholdEntry 8 }
```

The DS1 PM Day Threshold Table

```
-- This table contains 15 minute (current), and 24 hour threshold values
used
-- in performance parameter thresholding defined by ANSI T1.231

ds1pmDayTresholdTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF Ds1pmDayTresholdEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "The DS1 PM DayTreshold table."
        ::= { ds1pm 6 }

ds1pmDayTresholdEntry OBJECT-TYPE
        SYNTAX  Ds1pmDayTresholdEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "An entry in the DS1 PM DayTreshold table."
        INDEX   { ds1pmDayTresholdInterfaceIndex }
        ::= { ds1pmDayTresholdTable 1 }
Ds1pmDayTresholdEntry ::=
        SEQUENCE {
               ds1pmDayTresholdInterfaceIndex
                 INTEGER,
               ds1pmThreshESLDay
                 INTEGER,
               ds1pmThreshCVPDay
                 INTEGER,
               ds1pmThreshESPDay
                 INTEGER,
               ds1pmThreshSESPDay
                 INTEGER,
               ds1pmThreshSASPDay
                 INTEGER,
               ds1pmThreshCSSPDay
```

```
                INTEGER,
              ds1pmThreshUASPDay
                INTEGER
              }

ds1pmDayTresholdInterfaceIndex OBJECT-TYPE
        SYNTAX  INTEGER (1..'7fffffff'h)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The index value which uniquely identifies  the
                interface  to which this entry is applica-
                ble.  The interface identified by a  particular
                value  of  this  index is the same interface as
                identified by the same value as a ds1pmLineIndex
                object instance."
        ::= {   ds1pmDayTresholdEntry 1 }

ds1pmThreshESLDay OBJECT-TYPE
        SYNTAX  INTEGER (1..65535)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Line errored seconds 24 Hour (day) threshold."
        ::= { ds1pmDayTresholdEntry 2 }

ds1pmThreshCVPDay OBJECT-TYPE
        SYNTAX  INTEGER (1..1048575)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Path code violations 24 Hour (day) threshold."
        ::= { ds1pmDayTresholdEntry 3 }

ds1pmThreshESPDay OBJECT-TYPE
        SYNTAX  INTEGER (1..65535)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Path errored seconds 24 Hour (day) threshold."
        ::= { ds1pmDayTresholdEntry 4 }

ds1pmThreshSESPDay OBJECT-TYPE
        SYNTAX  INTEGER (1..4095)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Path severely errored seconds 24 Hour (day) threshold."
        ::= { ds1pmDayTresholdEntry 5 }

ds1pmThreshSASPDay OBJECT-TYPE
        SYNTAX  INTEGER (1..4095)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Path SEF/AIS seconds 24 Hour (day) threshold."
        ::= { ds1pmDayTresholdEntry 6 }

ds1pmThreshCSSPDay OBJECT-TYPE
        SYNTAX  INTEGER (1..4095)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Path controlled slip seconds 24 Hour (day) threshold."
        ::= { ds1pmDayTresholdEntry 7 }

ds1pmThreshUASPDay OBJECT-TYPE
        SYNTAX  INTEGER (1..4095)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Path unavailable seconds 24 Hour (day) threshold."
        ::= { ds1pmDayTresholdEntry 8 }

--      The cascadepm.ds3SuppMIB(2) branch is defined in the Cascade DS3
        Supplemental MIB.
--      Cascade DS3 Supplemental MIB which defines DS3/E3 PM branch is
        located
--      in cascadeds3.mib2 file in the same sub-directory as cascade.mib.
--      This MIB consists of delta objects that supplement the standard
        SONET MIB to
--      support SONET Performance Monitoring.  These supplements include:
--      o  ANSI T1.231 Support - increased set of counters, thresholds,
        total
--      24 hour counters, valid interval flags for current, 15 minute
        interval,
--      and 24 hour interval totals.
--      Consists of the following tables
--      The Configuration Table
--      Current Table
--      Interval Table
--      Total Table
--  Threshold Table
```

SONET PM Delta Configuration Table

```
sonetpmConfigTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF SonetpmConfigEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
               "The SONET PM Configuration table."
        ::= { sonetpm 1 }

sonetpmConfigEntry OBJECT-TYPE
        SYNTAX  SonetpmConfigEntry
        ACCESS  not-accessible
        STATUS  mandatory
```

```
        DESCRIPTION
                "An entry in the SONET PM Configuration table."
        INDEX   { sonetpmConfigInterfaceIndex }
        ::= { sonetpmConfigTable 1 }

SonetpmConfigEntry ::=
        SEQUENCE {
            sonetpmConfigInterfaceIndex
                 INTEGER,
            sonetpmConfigValidTotals
                 INTEGER,
            sonetpmConfigResetCurrent
                 INTEGER,
            sonetpmConfigResetInterval
                 INTEGER,
            sonetpmConfigResetTotal
                 INTEGER,
            sonetpmConfigThresholdCrossingEnable
                 INTEGER,
            sonetpmConfigSESThresholdSet
                 INTEGER,
            sonetpmConfigAlarmSoakTime
                 INTEGER,
            sonetpmConfigAlarmClearTime
                 INTEGER
               }
sonetpmConfigInterfaceIndex OBJECT-TYPE
        SYNTAX  INTEGER (1..'7fffffff'h)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This object is the identifier of an Inter-
                face on a managed device.  If there is an ifEn-
                try that is directly associated with  this  and
                only  this interface,  it should have the
                same value as ifIndex.   Otherwise,  the  value
                exceeds  ifNumber,  and  is a unique identifier
                following this rule: inside  interfaces  (e.g.,
                equipment  side)  with even numbers and outside
                interfaces  (e.g.,  network   side)   with   odd
                numbers."
        ::= { sonetpmConfigEntry 1 }

sonetpmConfigValidTotals OBJECT-TYPE
        SYNTAX  INTEGER (0..3)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of  previous  total intervals  for  which
                valid data was collected.  The value will be 3
                unless the interface was brought on-line within
                the last 3 days, in which case the value will
                be the number of complete 24 hour  intervals
                the since interface has been online."
        ::= { sonetpmConfigEntry 2 }

sonetpmConfigResetCurrent OBJECT-TYPE
        SYNTAX  INTEGER {
                   noReset (1),
                   resetCurrent (2)
                        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Reset the Current accumulation registers to 0."
        ::= { sonetpmConfigEntry 3 }

sonetpmConfigResetInterval OBJECT-TYPE
        SYNTAX  INTEGER (0..96)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Reset the Interval accumulation registers to 0.  The
                 number  of the interval to be cleared is given.  A
                 selection of 0 clears the entire interval table."
        ::= { sonetpmConfigEntry 4 }

sonetpmConfigResetTotal OBJECT-TYPE
        SYNTAX  INTEGER (0..3)
        ACCESS read-write
        STATUS  mandatory
        DESCRIPTION
                "Reset the Total accumulation registers to 0.  The
               interval number of the interval to be cleared is given.
               A selection of 0 clears the entire total table."
        ::= { sonetpmConfigEntry 5 }
```

```
sonetpmConfigThresholdCrossingEnable OBJECT-TYPE
        SYNTAX  INTEGER {
               disabled (1),
               enabled (2)
                        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Enable or disabled the detection and emission of
               threshold crossing alerts."
        ::= { sonetpmConfigEntry 6 }

sonetpmConfigSESThresholdSet OBJECT-TYPE
        SYNTAX  INTEGER {
               other (1),
               bellcore_tr_nwt_253_1991 (2),
               ansi_t1231_1993 (3),
               itu_g826_1995 (4)
                        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "SES and UAS threshold settings.  The setting determines
               which standard is used for SES and UAS thresholds.
               bellcore_tr_nwt_253_1991 - refers to Bellcore TR-NWT-
               00253, 1991 or ANSI T1M1.3/93-005R2,1993 ansi_t1231_1993 -
               refers to ANSI T1.231, 1993 or Bellcore GR-253-CORE, Issue
               2, 1995 itu_g826_1995 -refers to ITU recommendation G.826,
               1995"
        ::= { sonetpmConfigEntry 7 }

sonetpmConfigAlarmSoakTime OBJECT-TYPE
        SYNTAX  INTEGER (0..65535)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "This is the soak time for configurable alarms. An alarm
                 of this type must persist for this period before it is
                 declared.   The time is specified in millseconds units."
        ::= { sonetpmConfigEntry 8 }

sonetpmConfigAlarmClearTime OBJECT-TYPE
        SYNTAX  INTEGER (0..65535)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
               "This is the soak time for configurable alarms. An alarm
               of this type must remain clear for this period before it
               is declared to be cleared.   The time is specified in
               millseconds units."
        ::= { sonetpmConfigEntry 9 }
```

The SONET PM Delta Current Table

```
-- The table contains various statistics being
-- collected for the current 15 minute interval.

sonetpmCurrentTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF SonetpmCurrentEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "The SONET PM Current table."
        ::= { sonetpm 2 }

sonetpmCurrentEntry OBJECT-TYPE

        SYNTAX  SonetpmCurrentEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "An entry in the SONET PM Current table."
        INDEX   { sonetpmCurrentInterfaceIndex }
        ::= { sonetpmCurrentTable 1 }
```

```
SonetpmCurrentEntry ::=
        SEQUENCE {
             sonetpmCurrentInterfaceIndex
                 INTEGER,
             sonetpmCurrentInvalid
                 INTEGER,
             sonetpmCurrentCVLFE
                 Gauge,
             sonetpmCurrentAISSL
                 Gauge,
             sonetpmCurrentCVPFE
                 Gauge,
             sonetpmCurrentFCL
                 Gauge,
             sonetpmCurrentFCLFE
                 Gauge,
             sonetpmCurrentFCP
                 Gauge,
             sonetpmCurrentFCPFE
                 Gauge,
             sonetpmCurrentPSCL
                 Gauge,
             sonetpmCurrentESBS
                 Gauge,
             sonetpmCurrentESBL
                 Gauge,
             sonetpmCurrentESBP
                 Gauge,
             sonetpmCurrentPSDL
                 INTEGER
               }

sonetpmCurrentInterfaceIndex OBJECT-TYPE
        SYNTAX  INTEGER (1..'7fffffff'h)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The index value which uniquely identifies  the
                interface  to which this entry is applica-
                ble.  The interface identified by a  particular
                value  of  this  index is the same interface as
                identified by the same value as a ansi231LineIndex
                object instance."
        ::= { sonetpmCurrentEntry 1 }

sonetpmCurrentInvalid OBJECT-TYPE
        SYNTAX  INTEGER {
                data_invalid (1),
                data_valid (2)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current data invalid indicator."
        ::= { sonetpmCurrentEntry 2 }

sonetpmCurrentCVLFE OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current far-end FEBE-L.  FEBE-L is sent by
                 the far-end to indicate line BIP errors
                 have been detected at the far-end receiver."
        ::= { sonetpmCurrentEntry 3 }
```

```
sonetpmCurrentAISSL OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current line AIS defect seconds count.  A
                 second in which AIS-L has been detected."
        ::= { sonetpmCurrentEntry 4 }

sonetpmCurrentCVPFE OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current far-end FEBE-P count. FEBE-P is sent
                 by the far-end to indicate path BIP errors
                 have been detected at the far-end receiver."
        ::= { sonetpmCurrentEntry 5 }

sonetpmCurrentFCL OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current line failure count.  Count of AIS-L
                 failure events."
        ::= { sonetpmCurrentEntry 6 }

sonetpmCurrentFCLFE OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current far-end line failure count.  Count of
                 RFI-L events."
        ::= { sonetpmCurrentEntry 7 }

sonetpmCurrentFCP OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current path failure count.  Count of LOP-P or
                 AIS-P events."
        ::= { sonetpmCurrentEntry 8 }

sonetpmCurrentFCPFE OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current far-end path failure count.  Count of
                 RFI-P events."
        ::= { sonetpmCurrentEntry 9 }

sonetpmCurrentPSCL OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current Automatic Protection Switching event
                 count."
        ::= { sonetpmCurrentEntry 10 }

sonetpmCurrentESBS OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current ESB-S count.  An ESB-S second is one in
                 which two or more section BIP errors occurred and
                 no SEF or LOS condition was detected."
        ::= { sonetpmCurrentEntry 11 }

sonetpmCurrentESBL OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current ESB-L count.  An ESB-L second is one in
                 which two or more line BIP errors occurred and
                 no AIS-L occurred."
        ::= { sonetpmCurrentEntry 12 }

sonetpmCurrentESBP OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current ESB-P count.  An ESB-P second is one in
                 which two or more path BIP errors occurred and
                 no AIS-P or LOP occurred."
        ::= { sonetpmCurrentEntry 13 }

sonetpmCurrentPSDL OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Current Automatic Protection Switching duration.
                 The number of seconds a protection line is in
                 service due to protection switch events or the
                 number of seconds a working line is out-of-service
                 due to protection switch events."
        ::= { sonetpmCurrentEntry 14 }
```

The SONET PM Delta Interval Table

```
-- This table consists of entries that store history data for 96
-- 15-minute intervals
sonetpmIntervalTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF SonetpmIntervalEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "The SONET PM Interval table."
        ::= { sonetpm 3 }

sonetpmIntervalEntry OBJECT-TYPE
        SYNTAX   SonetpmIntervalEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "An entry in the SONET PM Interval table."
        INDEX   {  sonetpmIntervalInterfaceIndex, sonetpmIntervalNumber }
        ::= {  sonetpmIntervalTable 1 }

SonetpmIntervalEntry ::=
        SEQUENCE {
             sonetpmIntervalInterfaceIndex
                 INTEGER,
             sonetpmIntervalNumber
                 INTEGER,
             sonetpmIntervalInvalid
                 INTEGER,
             sonetpmIntervalCVLFE
                 Gauge,
             sonetpmIntervalAISSL
                 Gauge,
             sonetpmIntervalCVPFE
                 Gauge,
             sonetpmIntervalFCL
                 Gauge,
             sonetpmIntervalFCLFE
                 Gauge,
             sonetpmIntervalFCP
                 Gauge,
             sonetpmIntervalFCPFE
                 Gauge,
             sonetpmIntervalPSCL
                 Gauge,
             sonetpmIntervalESBS
                 Gauge,
             sonetpmIntervalESBL
                 Gauge,
             sonetpmIntervalESBP
                 Gauge,
             sonetpmIntervalPSDL
                 INTEGER
               }

sonetpmIntervalInterfaceIndex OBJECT-TYPE
        SYNTAX  INTEGER (1..'7fffffff'h)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The index value which uniquely identifies  the
                interface  to which this entry is applicable."
        ::= {  sonetpmIntervalEntry 1 }

sonetpmIntervalNumber OBJECT-TYPE
        SYNTAX  INTEGER (1..96)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The interval index."
        ::= { sonetpmIntervalEntry 2 }

sonetpmIntervalInvalid OBJECT-TYPE
        SYNTAX  INTEGER {
               data_invalid (1),
               data_valid (2)
                  }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Interval data invalid status."
        ::= { sonetpmIntervalEntry 3 }

sonetpmIntervalCVLFE OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Interval far-end FEBE-L count."
        ::= { sonetpmIntervalEntry 4 }

sonetpmIntervalAISSL OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Interval line AIS defect seconds count."
        ::= { sonetpmIntervalEntry 5 }

sonetpmIntervalCVPFE OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Interval far-end path FEBE count."
        ::= { sonetpmIntervalEntry 6 }
sonetpmIntervalFCL OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
```

```
        STATUS  mandatory
        DESCRIPTION
                "Interval line failure count."
        ::= { sonetpmIntervalEntry 7 }

sonetpmIntervalFCLFE OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Interval far-end line failure, RFI-L, count."
        ::= { sonetpmIntervalEntry 8 }

sonetpmIntervalFCP OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Interval path failure, LOP-P or AIS-P, count."
        ::= { sonetpmIntervalEntry 9 }

sonetpmIntervalFCPFE OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Interval far-end path failure, RFI-P, count."
        ::= { sonetpmIntervalEntry 10 }

sonetpmIntervalPSCL OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Interval Automatic Protection Switching event
                 count."
        ::= { sonetpmIntervalEntry 11 }

sonetpmIntervalESBS OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Interval ESB-S count."
        ::= { sonetpmIntervalEntry 12 }

sonetpmIntervalESBL OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Interval ESB-L count."
        ::= { sonetpmIntervalEntry 13 }

sonetpmIntervalESBP OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Interval ESB-P count."
        ::= { sonetpmIntervalEntry 14 }

sonetpmIntervalPSDL OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Interval Automatic Protection Switch duration."
        ::= { sonetpmIntervalEntry 15 }
```

The SONET PM Delta Total Table

```
-- This table contains 24 hour history registers for the last three
-- days: current, previous, and recent

sonetpmTotalTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF SonetpmTotalEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "The SONET PM total table."
        ::= { sonetpm 4 }

sonetpmTotalEntry OBJECT-TYPE
        SYNTAX  SonetpmTotalEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "An entry in the SONET PM total table."
        INDEX   { sonetpmTotalInterfaceIndex, sonetpmTotalIntervalNumber }
        ::= { sonetpmTotalTable 1 }
```

```
SonetpmTotalEntry ::=
        SEQUENCE {
            sonetpmTotalInterfaceIndex
                 INTEGER,
            sonetpmTotalIntervalNumber
                 INTEGER,
            sonetpmTotalIntervalInvalid
                 INTEGER,
            sonetpmTotalESS
                 Gauge,
            sonetpmTotalSESS
                 Gauge,
            sonetpmTotalSEFSS
                 Gauge,
            sonetpmTotalCVS
                 Gauge,
            sonetpmTotalESL
                 Gauge,
            sonetpmTotalSESL
                 Gauge,
            sonetpmTotalCVL
                 Gauge,
            sonetpmTotalUASL
                 Gauge,
            sonetpmTotalESP
                 Gauge,
            sonetpmTotalSESP
                 Gauge,
            sonetpmTotalCVP
                 Gauge,
            sonetpmTotalUASP
                 Gauge,
            sonetpmTotalCVLFE
                 Gauge,
            sonetpmTotalAISSL
                 Gauge,
            sonetpmTotalCVPFE
                 Gauge,
            sonetpmTotalFCL
                 Gauge,
            sonetpmTotalFCLFE
                 Gauge,
            sonetpmTotalFCP
                 Gauge,
            sonetpmTotalFCPFE
                 Gauge,
            sonetpmTotalPSCL
                 Gauge,
            sonetpmTotalESBS
                 Gauge,
            sonetpmTotalESBL
                 Gauge,
            sonetpmTotalESBP
                 Gauge,
            sonetpmTotalPSDL
                 INTEGER
        }

sonetpmTotalInterfaceIndex OBJECT-TYPE
        SYNTAX  INTEGER (1..'7fffffff'h)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The index value which uniquely identifies  the
                interface  to which this entry is applica-
                ble."
        ::= { sonetpmTotalEntry 1 }

sonetpmTotalIntervalNumber OBJECT-TYPE
        SYNTAX  INTEGER (1..3)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total table index."
        ::= { sonetpmTotalEntry 2 }

sonetpmTotalIntervalInvalid OBJECT-TYPE
        SYNTAX  INTEGER {
               data_invalid (1),
               data_valid (2)
               }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Interval data invalid status."
        ::= { sonetpmTotalEntry 3 }

sonetpmTotalESS OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total ES-S count.   An ES-S second is one
                 in which section BIP errors are present,
                 SEF has occurred, or LOS has occurred."
        ::= { sonetpmTotalEntry 4 }

sonetpmTotalSESS OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total SES-S count.  An SES-S second is one
                 in which the section BIP errors exceed the
                 SES threshold x, SEF has occurred, or LOS has
                 occurred."
        ::= { sonetpmTotalEntry 5 }
```

```
sonetpmTotalSEFSS OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total SEFS-S events.  An SEFS-S event is
                 one in which SEF (an SEF is declared when
                 4 contiguous frames have invalid frame
                 alignment words, an out-of-frame condition)."
        ::= { sonetpmTotalEntry 6 }

sonetpmTotalCVS OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total CV-S events.  CV-S count is the count of
                 section BIP errors detected."
        ::= { sonetpmTotalEntry 7 }

sonetpmTotalESL OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total ES-L count.  An ES-L second is one in
                 which line BIP errors are present, or AIS-L
                 has occurred."
        ::= { sonetpmTotalEntry 8 }

sonetpmTotalSESL OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total SES-L count.  An SES-L second is one
                 in which the line BIP errors exceed the
                 line SES threshold x, or AIS-L has occurred."
        ::= { sonetpmTotalEntry 9 }

sonetpmTotalCVL OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total CV-L count.  CV-L count is the count of
                 line BIP errors."
        ::= { sonetpmTotalEntry 10 }

sonetpmTotalUASL OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total UAS-L seconds count.  A line unavailable
                 second begins at the onset of 10 consecutive
                 SES-L seconds."
        ::= { sonetpmTotalEntry 11 }

sonetpmTotalESP OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total ES-P seconds.  An ES-P second is one in
                 which path BIP errors are present, LOP-P has
                 occurred, or AIS-P has occurred."
        ::= { sonetpmTotalEntry 12 }

sonetpmTotalSESP OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total SES-P seconds count.  An SES-P second is
                 one in which the path BIP errors exceed the
                 path SES threshold x, AIS-P has occurred, or
                 LOP-P has occurred."
        ::= { sonetpmTotalEntry 13 }

sonetpmTotalCVP OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total CV-P count.  CV-P count is the number
                 of path BIP errors."
        ::= { sonetpmTotalEntry 14 }

sonetpmTotalUASP OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total UAS-P seconds count.  A path unavailable
                 second begins at the onset of 10 consecutive
                 SES-P seconds."
        ::= { sonetpmTotalEntry 15 }
```

```
sonetpmTotalCVLFE OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total far-end FEBE-L count.  FEBE-L is
                 described under Current table."
        ::= { sonetpmTotalEntry 16 }

sonetpmTotalAISSL OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total AISS-L count.  AISS-L is described
                 under Current table."
        ::= { sonetpmTotalEntry 17 }

sonetpmTotalCVPFE OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total far-end FEBE-P count.  FEBE-P is
                 described under Current table."
        ::= { sonetpmTotalEntry 18 }

sonetpmTotalFCL OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total line failure count.  Described under
                 Current table."
        ::= { sonetpmTotalEntry 19 }

sonetpmTotalFCLFE OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total far-end failure count.  Described under
                 Current table."
        ::= { sonetpmTotalEntry 20 }

sonetpmTotalFCP OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total path failure count.  Described under
                 Current table."
        ::= { sonetpmTotalEntry 21 }

sonetpmTotalFCPFE OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total far-end path failure count.  Described
                 under Current table."
        ::= { sonetpmTotalEntry 22 }

sonetpmTotalPSCL OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total Automatic Protection Switching event
                 count."
        ::= { sonetpmTotalEntry 23 }

sonetpmTotalESBS OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total ESB-S count.  Described under Current
                 table."
        ::= { sonetpmTotalEntry 24 }

sonetpmTotalESBL OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total ESB-L count.  Described under Current
                 table."
        ::= { sonetpmTotalEntry 25 }

sonetpmTotalESBP OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total ESB-P count.  Described under Current
                 table."
        ::= { sonetpmTotalEntry 26 }

sonetpmTotalPSDL OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Total Automatic Protection Switching duration."
        ::= { sonetpmTotalEntry 27 }
```

The SONET PM Threshold Table

```
-- This table contains 15 minute (current), and 24 hour threshold values
used
-- in performance parameter thresholding defined by ANSI T1.231

sonetpmThresholdTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF SonetpmThresholdEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "The SONET PM Threshold table."
        ::= { sonetpm 5 }

sonetpmThresholdEntry OBJECT-TYPE
        SYNTAX  SonetpmThresholdEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "An entry in the SONET PM Threshold table."
        INDEX   { sonetpmThresholdInterfaceIndex }
        ::= { sonetpmThresholdTable 1 }

SonetpmThresholdEntry ::=
        SEQUENCE {
               sonetpmThresholdInterfaceIndex
                 INTEGER,
               sonetpmThreshCVSCurrent
                 INTEGER,
               sonetpmThreshCVSDay
                 INTEGER,
               sonetpmThreshESSCurrent
                 INTEGER,
               sonetpmThreshESSDay
                 INTEGER,
               sonetpmThreshSESSCurrent
                 INTEGER,
               sonetpmThreshSESSDay
                 INTEGER,
               sonetpmThreshCVLCurrent
                 INTEGER,
               sonetpmThreshCVLDay
                 INTEGER,
               sonetpmThreshESLCurrent
                 INTEGER,
               sonetpmThreshESLDay
                 INTEGER,
               sonetpmThreshSESLCurrent
                 INTEGER,
               sonetpmThreshSESLDay
                 INTEGER,
               sonetpmThreshUASLCurrent
                 INTEGER,
               sonetpmThreshUASLDay
                 INTEGER,
               sonetpmThreshCVPCurrent
                 INTEGER,
               sonetpmThreshCVPDay
                 INTEGER,
               sonetpmThreshESPCurrent
                 INTEGER,
               sonetpmThreshESPDay
                 INTEGER,
               sonetpmThreshSESPCurrent
                 INTEGER,
               sonetpmThreshSESPDay
                 INTEGER,
               sonetpmThreshUASPCurrent
                 INTEGER,
               sonetpmThreshUASPDay
                 INTEGER
               }

sonetpmThresholdInterfaceIndex OBJECT-TYPE
        SYNTAX  INTEGER (1..'7fffffff'h)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The index value which uniquely identifies  the
                interface  to which this entry is
               applicable."
        ::= {   sonetpmThresholdEntry 1 }

sonetpmThreshCVSCurrent OBJECT-TYPE
        SYNTAX  INTEGER (1..16383)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Current section CV threshold."
        ::= { sonetpmThresholdEntry 2 }

sonetpmThreshCVSDay OBJECT-TYPE
        SYNTAX  INTEGER (1..1048575)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Day total section CV threshold."
        ::= { sonetpmThresholdEntry 3 }
```

```
sonetpmThreshESSCurrent OBJECT-TYPE
        SYNTAX  INTEGER (1..900)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Current section ES threshold."
        ::= { sonetpmThresholdEntry 4 }

sonetpmThreshESSDay OBJECT-TYPE
        SYNTAX  INTEGER (1..65535)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Day total ES threshold."
        ::= { sonetpmThresholdEntry 5 }

sonetpmThreshSESSCurrent OBJECT-TYPE
        SYNTAX  INTEGER (1..63)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Current section SES threshold."
        ::= { sonetpmThresholdEntry 6 }

sonetpmThreshSESSDay OBJECT-TYPE
        SYNTAX  INTEGER (1..4095)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Day total SES threshold."
        ::= { sonetpmThresholdEntry 7 }

sonetpmThreshCVLCurrent OBJECT-TYPE
        SYNTAX  INTEGER (1..16383)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Current line CV threshold."
        ::= { sonetpmThresholdEntry 8 }

sonetpmThreshCVLDay OBJECT-TYPE
        SYNTAX  INTEGER (1..1048575)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Day total line CV threshold."
        ::= { sonetpmThresholdEntry 9 }

sonetpmThreshESLCurrent OBJECT-TYPE
        SYNTAX  INTEGER (1..900)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Current line ES threshold."
        ::= { sonetpmThresholdEntry 10 }

sonetpmThreshESLDay OBJECT-TYPE
        SYNTAX  INTEGER (1..65535)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Day total line ES threshold."
        ::= { sonetpmThresholdEntry 11 }

sonetpmThreshSESLCurrent OBJECT-TYPE
        SYNTAX  INTEGER (1..63)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Current line SES threshold."
        ::= { sonetpmThresholdEntry 12 }

sonetpmThreshSESLDay OBJECT-TYPE
        SYNTAX  INTEGER (1..4095)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Day total line SES threshold."
        ::= { sonetpmThresholdEntry 13 }

sonetpmThreshUASLCurrent OBJECT-TYPE
        SYNTAX  INTEGER (1..63)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Current line UAS threshold."
        ::= { sonetpmThresholdEntry 14 }

sonetpmThreshUASLDay OBJECT-TYPE
        SYNTAX  INTEGER (1..4095)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Day total line UAS threshold."
        ::= { sonetpmThresholdEntry 15 }
```

```
sonetpmThreshCVPCurrent OBJECT-TYPE
        SYNTAX  INTEGER (1..16383)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Current path CV threshold."
        ::= { sonetpmThresholdEntry 16 }

sonetpmThreshCVPDay OBJECT-TYPE
        SYNTAX  INTEGER (1..1048575)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Day total path CV threshold."
        ::= { sonetpmThresholdEntry 17 }

sonetpmThreshESPCurrent OBJECT-TYPE
        SYNTAX  INTEGER (1..900)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Current path ES threshold."
        ::= { sonetpmThresholdEntry 18 }

sonetpmThreshESPDay OBJECT-TYPE
        SYNTAX  INTEGER (1..65535)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Day total path ES threshold."

        ::= { sonetpmThresholdEntry 19 }

sonetpmThreshSESPCurrent OBJECT-TYPE
        SYNTAX  INTEGER (1..63)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Current path SES threshold."
        ::= { sonetpmThresholdEntry 20 }

sonetpmThreshSESPDay OBJECT-TYPE
        SYNTAX  INTEGER (1..4095)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Day total path SES threshold."
        ::= { sonetpmThresholdEntry 21 }

sonetpmThreshUASPCurrent OBJECT-TYPE
        SYNTAX  INTEGER (1..63)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Current path UAS threshold."
        ::= { sonetpmThresholdEntry 22 }

sonetpmThreshUASPDay OBJECT-TYPE
        SYNTAX  INTEGER (1..4095)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Day total path UAS threshold."
        ::= { sonetpmThresholdEntry 23 }
```

The ATM Accounting Group

```
--      Global objects, Lport-based objects, and Ckt-based objects

atmacctControl OBJECT-TYPE
        SYNTAX INTEGER {
                disabled   (1),
                pvcenabled (2),
                svcenabled (3),
                enabled    (4)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
  "             This object defines the capability to enable or disable
                usage-based ATM accounting at the switch level.

                The possible values are:
                disabled   - Usage measurement is disabled
                pvcenabled - Usage measurement is enabled for PVCs only
                svcenabled - Usage measurement is enabled for SVCs only
                enabled    - Usage measurement is enabled for PVCs and SVCs

                When the value of this object is a value other than
                'disabled', the value of a logical port's accounting
                capability objects will take precedence.

                When the value of this object is 'disabled', it overrides
                all logical ports' accounting capability objects and
                accounting is disabled across the entire switch.

                The default value of this object is 'disabled'.
                This object is considered to be to 'administrative' state
                of the ATM accounting system on the switch, whereas the
                object atmacctOperState is the corresponding operational
                state."
 ::= { atmacct 1 }
```

```
atmacctASAddressPri OBJECT-TYPE
        SYNTAX IpAddress
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The IP Address of the primary Accounting Server
                that is servicing ATM accounting for this switch."
        ::= { atmacct 2 }

atmacctSwASCommsFailures OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The number of times that communication from the switch to
                the ATM Accounting Server has failed during the current
                day.  A failure signifies failure of a file transfer
                operation to the Accounting Server."
        ::= { atmacct 3 }

atmacctPvcAggrPeriod OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object defines the recording interval over which PVC
                usage measurements are taken and transferred to the Adjunct
                Processor (Accounting Server), as defined by Bellcore
                GR-1110-CORE.

                Acceptable values represent 15-minute increments.  The
                minimum value is 1 (15 minutes).  The maximum value is 96
                (24 hours).

                The default value is 4 (1 hour)."
        ::= { atmacct 4 }

atmacctPvcCurAggrPeriodStart OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The start-time (GMT) of the current ATM PVC recording
                interval."
        ::= { atmacct 5 }

atmacctPvcCurAggrPeriodEnd OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The end-time (GMT) of the current ATM PVC recording
                interval."
        ::= { atmacct 6 }

atmacctCollectionPeriod OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object defines how often the switch snapshots the
                state of all ATM PVCs and SVCs to stable storage.

                Acceptable values represent 5-minute increments.  The
                minimum value is 0 (no snapshots).  The maximum value is
                12 (1 hour).

                The default value is 1 (5 minutes)."
        ::= { atmacct 7 }

atmacctSvcUsageRecCreated OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Count of the number of new SVCs for which usage records
                were created during the current rate period."
        ::= { atmacct 8 }

atmacctSvcTotalUsageRecCreated OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Count of the number of new SVCs for which usage records
                were created during the current day."
        ::= { atmacct 9 }

atmacctPvcUsageRecCreated OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "This object reports the number of PVC usage records that
                were created at the end of the previous recording interval.
                This counter indicates the number of PVCs for which usage
                measurement was enabled during the last recording
                interval."
        ::= { atmacct 10 }
```

```
atmacctUsageRecCrFailures OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Total number of usage records that could not be created
                during the current day."
        ::= { atmacct 11 }

atmacctUsageRecSent OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Total number of usage records that have been transferred
                to the ATM Accounting Server during the current day."
        ::= { atmacct 12 }

atmacctAdminAction OBJECT-TYPE
        SYNTAX INTEGER {
                invalid         (1),
                forceUpload     (2),
                resetBWstats    (3)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object defines a set of administrative actions that
                can be performed by the ATM Accounting System.
                forceUpload  - Forces an upload of any queued ATM usage
                data to the ATM Accounting Server.
                resetBWstats - Reset the AS Communications Bandwidth
                tracking statistics to zero.

                This object always returns invalid(1) when read.
  "
        ::= { atmacct 13 }

atmacctOamCellCounting OBJECT-TYPE
        SYNTAX INTEGER {
                dontcount       (1),
                includeintotal  (2)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object controls the recording of OAM cell when PVC
                and/or SVC usage measurement is enabled.

                Possible values are:
                dontcount   = Don't count OAM cells at all
                includeintotal  = Include OAM cell counts in total cell
                counts (Required by Bellcore GR-1110)

                The default value of this object is includeintotal(2).
                Note: this object is read-only unless PVC billing is
                disabled at the global level."
        ::= { atmacct 14 }

atmacctSvcCbrCellCounting OBJECT-TYPE
        SYNTAX INTEGER {
                disabled     (1),
                intraenabled (3)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object is used to enable or disable CBR SVC cell
                counting when CBR Recording is enabled.

                For Intra-network CBR SVCs, the switch will record CBR
                cell counts for the SVC if and only if the parameter is
                set to intraenabled(3) and CBR Recording is enabled (at the
                switch and logical port level).

                This parameter will also support CBR cell-counting for
                inter-network SVCs in the future.

                The default value of this object is intraenabled(3)."
        ::= { atmacct 15 }
```

```
atmacctSvcAbrRecording OBJECT-TYPE
        SYNTAX INTEGER {
                disabled     (1),
                intraenabled (3)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object is used to enable or disable the recording of
                ATM ABR SVCs.

                For Intra-network ABR SVCs, the switch will generate
                usage data for the SVC if and only if the parameter is set
                to intraenabled(3) and the corresponding logical port ABR
                Recording parameter is set to enabled or study.

                This parameter also support ABR recording for
                inter-network SVCs in the future.

                The default value of this object is intraenabled(3)."
        ::= { atmacct 16 }

atmacctSvcCbrRecording OBJECT-TYPE
        SYNTAX INTEGER {
                disabled     (1),
                intraenabled (3)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object is used to enable or disable the recording of
                ATM CBR SVCs.

                For Intra-network CBR SVCs, the switch will generate
                usage data for the SVC if and only if the parameter is set
                to intraenabled(3) and the corresponding logical port CBR
                Recording parameter is set to enabled or study.

                This parameter also support CBR recording for
                inter-network SVCs in the future.

                The default value of this object is intraenabled(3)."
        ::= { atmacct 17 }

atmacctSvcUbrRecording OBJECT-TYPE
        SYNTAX INTEGER {
                disabled     (1),
                intraenabled (3)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object is used to enable or disable the recording of
                ATM UBR SVCs.

                For Intra-network UBR SVCs, the switch will generate
                usage data for the SVC if and only if the parameter is set
                to intraenabled(3) and the corresponding logical port UBR
                Recording parameter is set to enabled or study.

                This parameter also support UBR recording for
                inter-network SVCs in the future.

                The default value of this object is intraenabled(3)."
        ::= { atmacct 18 }

atmacctSvcVbrRecording OBJECT-TYPE
        SYNTAX INTEGER {
        disabled     (1),
        intraenabled (3)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object is used to enable or disable the recording of
                ATM VBR SVCs.

                For Intra-network VBR SVCs, the switch will generate
                usage data for the SVC if and only if the parameter is set
                to intraenabled(3) and the corresponding logical port VBR
                Recording parameter is set to enabled or study.

                This parameter also support VBR recording for
                inter-network SVCs in the future.

                The default value of this object is intraenabled(3)."
        ::= { atmacct 19 }

atmacctAvgTransportBwUsed OBJECT-TYPE
        SYNTAX Gauge
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "This object reports the average amount of bandwidth (in
                bits per second) that has been used to transport ATM
                Accounting data to the Accounting Server during the current
                day."
        ::= { atmacct 20 }
```

```
atmacctAvgTransportBwBurst OBJECT-TYPE
        SYNTAX Gauge
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "This object reports the average transport bandwidth burst
                rate (in bits per second) obtained to transport ATM
                accounting data to the Accounting Server during the
                current day."
        ::= { atmacct 21 }

atmacctMinTransportBwBurst OBJECT-TYPE
        SYNTAX Gauge
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "This object reports the minimum transport bandwidth burst
                rate (in bits per second) obtained to transport ATM
                accounting data to the Accounting Server during the
                current day."
        ::= { atmacct 22 }

atmacctMaxTransportBwBurst OBJECT-TYPE
        SYNTAX Gauge
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "This object reports the maximum transport bandwidth burst
                rate (in bits per second) obtained to transport ATM
                accounting data to the Accounting Server during the
                current day."
 ::= { atmacct 23 }

atmacctLportTable OBJECT-TYPE
        SYNTAX SEQUENCE OF AtmacctLportEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "A table of lport-related ATM Accounting System managable
                objects, indexed by logical port identifier."
        ::= { atmacct 24 }

atmacctLportEntry OBJECT-TYPE
        SYNTAX AtmacctLportEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "An atmacctLportEntry contains a set of lport-related
                ATM Accounting System managable objects, indexed by logical
                port identifier."
        INDEX { atmacctLportIfIndex }

        ::= { atmacctLportTable 1 }

AtmacctLportEntry ::=
        SEQUENCE {
                atmacctLportIfIndex
                    Index,
                atmacctLportSvcControl
                    INTEGER,
                atmacctLportSvcPtPtRecording
                    INTEGER,
                atmacctLportSvcPtMPtRecording
                    INTEGER,
                atmacctLportSvcDefaultAddress
                    OCTET STRING,
                atmacctLportSvcIntraAbrRecording
                    INTEGER,
                atmacctLportSvcIntraCbrRecording
                    INTEGER,
                atmacctLportSvcIntraUbrRecording
                    INTEGER,
                atmacctLportSvcIntraVbrRecording
                    INTEGER,
                atmacctLportSvcUnsuccRecording
                    INTEGER,
                atmacctLportSvcSubAddressRecording
                    INTEGER,
                atmacctLportPvcCarrierId
                    INTEGER,
                atmacctLportPvcParamRecording
                    INTEGER,
                atmacctLportSvcDefaultAddressType
                    INTEGER
                }

atmacctLportIfIndex OBJECT-TYPE
        SYNTAX Index
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
  "             Index: logical-port identifier."
        ::= { atmacctLportEntry 1 }
```

```
atmacctLportSvcControl OBJECT-TYPE
        SYNTAX INTEGER {
                disabled (1),
                enabled  (2)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object defines the ability to enable and disable ATM
                SVC accounting on this logical port.  When the value of
                atmacctControl is 'enabled', the value of this object will
                take precedence.  When the value of atmacctControl is
                'disabled', the value of this object will be overridden and
                ATM SVC accounting will be disabled.

                ‘The default value of this object is 'disabled'.
  "
        ::= { atmacctLportEntry 2 }

atmacctLportSvcPtPtRecording OBJECT-TYPE
        SYNTAX INTEGER {
                disabled    (1),
                originating (2),
                terminating (3),
                enabled     (4)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object is used to enable or disable the recording of
                usage information for Point to Point ATM SVCs that
                originate or terminate on this port.
                disabled    = Usage measurement will not be performed for
                SVC calls on this port.
                originating = Usage data is generated only for calls that
                originate on this port.
                terminating = Usage data is generated only for calls that
                terminate on this port.
                enabled     = Usage measurement will be performed at both
                originating and terminating ends of all calls
                on this port.

                The default value of this object is enabled(4).

                When set to a value other than disabled(1), unsuccessful
                ATM
                VC calls will be recorded according to the value of the
                atmacctLportUnsuccSvcRecording object.  Otherwise
                unsuccessful calls will not be recorded."

        ::= { atmacctLportEntry 3 }

atmacctLportSvcPtMPtRecording OBJECT-TYPE
        SYNTAX INTEGER {
                disabled    (1),
                originating (2),
                terminating (3),
                enabled     (4)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object is used to enable or disable the recording of
                usage information for Point to Multi-Point ATM SVCs that
                originate or terminate on this port.

                disabled    = Usage measurement will not be performed for
                SVC calls on this port.
                originating = Usage data is generated only for calls that
                originate on this port.
                terminating = Usage data is generated only for calls that
                terminate on this port.
                enabled     = Usage measurement will be performed at both
                originating and terminating ends of all calls
                on this port.

                The default value of this object is terminating(3)

                When set to a value other than disabled(1), unsuccessful
                ATM SVC calls will be recorded according to the value of
                the atmacctLportUnsuccSvcRecording object.  Otherwise
                unsuccessful calls will not be recorded."
        ::= { atmacctLportEntry 4 }

atmacctLportSvcDefaultAddress OBJECT-TYPE
        SYNTAX OCTET STRING(SIZE(1..20))
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object defines a Default Billing Address for ATM SVCs
                which originate at his lport.  Note that this address may
                be different from the Default UNI Address defined for
                Calling Party Insertion.

                The Default address for the port must be recorded at the
                switch whenever:
                1) no Calling Party Number is present, or
                2) the Calling Party Number fails screening or is invalid,
                or
                3) the Calling Party Number is different from the default
                address."
        ::= { atmacctLportEntry 5 }
```

```
atmacctLportSvcIntraAbrRecording OBJECT-TYPE
        SYNTAX INTEGER {
                disabled (1),
                enabled  (2),
                tudy     (3)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object is used to enable or disable the recording of
                Intranetwork ATM ABR SVCs at the UNI.  The switch will
                generate usage data for the SVC only if the parameter
                is set to enabled(2) or study(3).

                If this parameter is set to study, usage data that is
                generated is marked as 'study', per Bellcore GR-1110-CORE.

                This parameter does not apply to ABR recording for
                inter-network SVCs.

                The default value of this object is enabled(2)."
                ::= { atmacctLportEntry 6 }

atmacctLportSvcIntraCbrRecording OBJECT-TYPE
        SYNTAX INTEGER {
                disabled (1),
                enabled  (2),
                study    (3)
   }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object is used to enable or disable the recording of
                Intranetwork ATM CBR SVCs at the UNI.  The switch will
                generate usage data for the SVC only if the parameter
                is set to enabled(2) or study(3).

                If this parameter is set to study, usage data that is
                generated is marked as 'study', per Bellcore GR-1110-CORE.

                This parameter does not apply to CBR recording for
                inter-network SVCs.

                The default value of this object is enabled(2)."
        ::= { atmacctLportEntry 7 }

atmacctLportSvcIntraUbrRecording OBJECT-TYPE
        SYNTAX INTEGER {
                disabled (1),
                enabled  (2),
                study    (3)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object is used to enable or disable the recording of
                Intranetwork ATM UBR SVCs at the UNI.  The switch will
                generate usage data for the SVC only if the parameter
                is set to enabled(2) or study(3).

                If this parameter is set to study, usage data that is
                generated is marked as 'study', per Bellcore GR-1110-CORE.

                This parameter does not apply to UBR recording for
                inter-network SVCs.

                The default value of this object is enabled(2)."
        ::= { atmacctLportEntry 8 }

atmacctLportSvcIntraVbrRecording OBJECT-TYPE
        SYNTAX INTEGER {
                disabled (1),
                enabled  (2),
                study    (3)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object is used to enable or disable the recording of
                Intranetwork ATM VBR SVCs at the UNI.  The switch will
                generate usage data for the SVC only if the parameter
                is set to enabled(2) or study(3).

                If this parameter is set to study, usage data that is
                generated is marked as 'study', per Bellcore GR-1110-CORE.

                This parameter does not apply to VBR recording for
                inter-network SVCs.

                The default value of this object is enabled(2)."
        ::= { atmacctLportEntry 9 }
```

```
atmacctLportSvcUnsuccRecording OBJECT-TYPE
        SYNTAX INTEGER {
                disabled    (1),
                originating (2),
                terminating (3),
                enabled     (4)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object is used to enable or disable the recording of
                usage information for unsuccessful ATM SVCs that originate
                or terminate on this port.  Note that this parameter
                governs recording for both point-to-point and point-to
                -multipoint SVCs.

                disabled    = Usage data will not be generated for
                unsuccessful calls on this port.
                originating = Usage data is generated for all unsuccessful
                calls that originated on this port.  This
                value can only be set for UNI ports.
                terminating = Usage data is generated for all unsuccessful
                calls that terminated on this port.
                enabled     = Usage data is generated for all unsuccessful
                calls that originated or terminated on this
                port.  This value can only be set for UNI ports.

                The default value of this object is enabled(4) at the UNI
                and disabled(1) at the network interface.

                The acceptable values of this object on a network interface
                are disabled(1) or terminating(3)."
        ::= { atmacctLportEntry 10 }

atmacctLportSvcSubAddressRecording OBJECT-TYPE
        SYNTAX INTEGER {
                disabled     (1),
                callingParty (2),
                calledParty  (3),
                enabled      (4)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object is used to enable or disable the recording of
                the Sub-addresses in ATM SVC accounting records at the UNI.

                disabled     = Do not records Sub-addresses
                callingParty = Record the Calling Party Sub-address when
                present in a call
                calledParty  = Record the Called Party Sub-address when
                present in a call
                enabled      = Record both sub-addresses when present
                        The default value of this object is disabled(1)."
        ::= { atmacctLportEntry 11 }

atmacctLportPvcCarrierId OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object is a 5 digit decimal number that is used to
                identify the interconnected carrier associated with ATM
                PVCs at a network interface (B-ICI).

                Note: this object is read-only if PVCs are provisioned on
                the specified port."
        ::= { atmacctLportEntry 12 }

atmacctLportPvcParamRecording OBJECT-TYPE
        SYNTAX INTEGER {
                disabled (1),
                enabled  (2)
                }
 ACCESS read-write
 STATUS mandatory
 DESCRIPTION
        "This object controls the recording of ATM PVC parameters
        when usage measurement is enabled for this port.  When set
        to enabled(2), all of the following parameters (if defined
        for the circuit) will be recorded in the usage data for
        each PVC:
        Ingress Peak Cell Rates (CLP=0) and (CLP=0+1)
        Ingress Sustained Cell Rates (CLP=0) and (CLP=0+1)
        Ingress QoS Class
        Egress Peak Cell Rates (CLP=0) and (CLP=0+1)
        Egress Sustained Cell Rates (CLP=0) and (CLP=0+1)
                Egress QoS Class

                This object is defined at the user (UNI) and network
                interfaces (B-ICI and NNI).

                The default value of this object is disabled(1).

                Note: this object is read-only if PVCs are provisioned on
                the specified port."
        ::= { atmacctLportEntry 13 }
```

```
atmacctLportSvcDefaultAddressType OBJECT-TYPE
        SYNTAX INTEGER {
                e164 (1),
                atm-endsystem (2),
                unknown (4),
                useCPIaddress (5)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object identifies the type of ATM address that is
                to be used as the default billing address for SVCs
                originating on this logical port.  The octet string
                comprising thisaddress is given by parameter
                atmacctLportSvcDefaultAddress.

                Note: if the value of this parameter is useCPIaddress (5),
                the switch will use the Calling Party Insertion address
                (object svcConfigCgPtyInsertionAddress) defined for this
                logical port as the default billing address.  In this case,
                the value of atmacctLportSvcDefaultAddress has no meaning,
                and should be set to null."
        ::= { atmacctLportEntry 14 }

atmacctCktTable OBJECT-TYPE
        SYNTAX SEQUENCE OF AtmacctCktEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "A table of ckt-related ATM Accounting System managable
                objects, indexed by source logical port id and source
                DLCI (concatenated VPI:VCI)."
        ::= { atmacct 25 }

atmacctCktEntry OBJECT-TYPE
        SYNTAX AtmacctCktEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "An atmacctCktEntry contains a set of ckt-related
                ATM Accounting System managable objects, indexed by source
                logical port id and source DLCI (concatenated VPI:VCI)."
        INDEX { atmacctCktSrcIfIndex, atmacctCktSrcDlci }
        ::= { atmacctCktTable 1 }

AtmacctCktEntry ::=
        SEQUENCE {
                atmacctCktSrcIfIndex
                    Index,
                atmacctCktSrcDlci
                    INTEGER,
                atmacctCktControl
                    INTEGER,
                atmacctCktUsageMeasurement
                    INTEGER,
                atmacctCktChargeablePartyId
                    OCTET STRING,
                atmacctCktSendNew
                    INTEGER,
                atmacctCktCreationTime
                    INTEGER
                }

atmacctCktSrcIfIndex OBJECT-TYPE
        SYNTAX Index
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Index: source logical-port identifier."
        ::= { atmacctCktEntry 1 }

atmacctCktSrcDlci OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Index: source DLCI (concatenated VPI:VCI)."
        ::= { atmacctCktEntry 2 }

atmacctCktControl OBJECT-TYPE
        SYNTAX INTEGER {
                disabled (1),
                enabled  (2),
                study    (3)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object is used to enable or disable ATM PVC recording
                for accounting purposes.

                disabled = The PVC will not be recorded at this interface
                enabled  = The PVC will be recorded at this interface
                study    = The PVC will be recorded and marked as study
                (as defined by Bellcore GR-1110-CORE)

                This object is defined at the user (UNI) and network
                interfaces (B-ICI and NNI).

                The default value of this object is enabled(2) at the UNI
                and enabled(2) at the network interface."
                ::= { atmacctCktEntry 3 }
```

```
atmacctCktUsageMeasurement OBJECT-TYPE
        SYNTAX INTEGER {
                disabled (1),
                egress   (2),
                ingress  (3),
                enabled  (4)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object is used to enable or disable the recording of
                usage counts (e.g., cell or frame counts) for an ATM PVC.

                disabled = No usage counts are generated for this circuit
                egress   = Egress usage counts are generated, where egress
                refers to data sent to the user (on a UNI) or to
                the network (on a network interface).
                ingress  = Ingress usage counts are generated, where
                ingress
                refers to data received from the user (on a UNI)
                or from the network (on a network interface).
                enabled  = Both ingress and egress counts are generated

                Note: usage records are not generated if PVC Recording
                (atmacctCktControl) is disabled.

                This object is defined at the user (UNI) and network
                interfaces (B-ICI and NNI).

                The default value of this object is disabled(1) at the UNI
                and disabled(1) at the network interface.

                Note: this object is read-only after initial provisioning
                of the circuit."
        ::= { atmacctCktEntry 4 }

atmacctCktChargeablePartyId OBJECT-TYPE
        SYNTAX OCTET STRING(SIZE(1..16))
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object defines a 1 to 16 digit decimal chargeable
                party for this ATM PVC, per Bellcore GR-1110-CORE.

                Note: this object is read-only after initial provisioning
                of the circuit."
        ::= { atmacctCktEntry 5 }

atmacctCktSendNew OBJECT-TYPE
        SYNTAX INTEGER {
                invalid (1),
                forceSend (2)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object provides an interface for the Accounting
                Server
                to request the NEW usage data record for the corresponding
                circuit.  Upon read, this object always returns
                invalid(1)."
        ::= { atmacctCktEntry 6 }

atmacctCktCreationTime OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object represents the time (UCT, seconds since
                Jan. 1, 1970) that the circuit was created.  Once created,
                this object is read-only, since it is used for correlating
                accounting records for the circuit."
        ::= { atmacctCktEntry 7 }

atmacctStressTestRate OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The rate (calls/sec/IOM) of simulated SVCs, for stress
                testing the accounting components on the IOMs and SP.

                This object is for internal debugging purposes, and is
                read-only in release versions."
        ::= { atmacct 26 }

atmacctASAddressSec OBJECT-TYPE
        SYNTAX IpAddress
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The IP Address of the secondary Accounting Processor that
                is servicing ATM accounting for this switch."
        ::= { atmacct 27 }
```

```
atmacctASControl OBJECT-TYPE
        SYNTAX INTEGER {
                primary (1),
                secondary (2)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object controls which configured ATM Accounting
                Accounting Server address is to be used for transferring
                usage data."
        ::= { atmacct 28 }

atmacctOperState OBJECT-TYPE
        SYNTAX INTEGER {
                disabled   (1),
                pvcenabled (2),
                svcenabled (3),
                enabled    (4)
                }
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "This object corresponds to the operational state of the
                ATM Accounting system on the switch.

                The possible values are:
                disabled   - Usage measurement is disabled
                pvcenabled - Usage measurement is enabled for PVCs only
                svcenabled - Usage measurement is enabled for SVCs only
                enabled    - Usage measurement is enabled for PVCs and SVCs

                This object is considered to be the 'operational' state
                of the ATM Accounting system on the switch, whereas the
                object
                atmacctControl is the corresponding 'administrative'
                state."
        ::= { atmacct 29 }

atmacctASCommsState OBJECT-TYPE
        SYNTAX INTEGER {
                red    (1),
                yellow (2),
                green  (3)
                }
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "This object identifies the current state of communications
                to the ATM Accounting Server.

                Possible values are:

                red    - File transfers have repeatedly failed, and all
                         switch resources for storing additional accounting
                         data have been exhausted.  The operational state of
                         ATM accounting for the switch has been downgraded
                         to DISABLED.
                         yellow - File transfers are experiencing significant
                         failures.  Several unsuccessful attempts have
                         been made to transport the file at the head of the
                         queue.  New data is continuing to be generated,
                         and queued up.  The operational state of ATM
                         Accounting has not been downgraded yet.
                         green  - File transfers are not experiencing significant
                         failures.  Any accounting data queued during
                         failure periods has been successfully transported
                         to the Accounting Server."
        ::= { atmacct 30 }

atmacctLastBWResetTime OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The last time (GMT) that the AS Communications Bandwidth
                tracking statistics were reset to zero."
        ::= { atmacct 31 }

-- cascatm branch
-- atmckt group
-- ctlCktTable

ctlCktTable OBJECT-TYPE
        SYNTAX SEQUENCE OF CtlCktEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "ATM Control Circuit table."
        ::= { atmckt 1 }

ctlCktEntry OBJECT-TYPE
        SYNTAX CtlCktEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "ATM Control Circuit Entry"
        INDEX { ctlCktifIndex, ctlCktVpi, ctlCktVci }
        ::= { ctlCktTable 1 }
```

```
CtlCktEntry ::=
        SEQUENCE {
                ctlCktifIndex
                        INTEGER,
                ctlCktVpi
                        INTEGER,
                ctlCktVci
                        INTEGER,
                ctlCktType
                        INTEGER,
                ctlCktInCells
                        Counter,
                ctlCktOutCells
                        Counter,
                ctlCktInDiscardedClp0Cells
                        Counter,
                ctlCktInDiscardedClp1Cells
                        Counter,
                ctlCktInPassedClp0Cells
                        Counter,
                ctlCktInPassedClp1Cells
                        Counter,
                ctlCktInTaggedCells
                        Counter,
                ctlCktOutClp0Cells
                        Counter,
                ctlCktOutClp1Cells
                        Counter
               }

ctlCktifIndex OBJECT-TYPE
        SYNTAX INTEGER (0..4095)
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Interface on which control channel exists"
        ::= { ctlCktEntry 1 }

ctlCktVpi OBJECT-TYPE
        SYNTAX INTEGER (0..4095)
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "VPI of control channel"
        ::= { ctlCktEntry 2 }

ctlCktVci OBJECT-TYPE
        SYNTAX INTEGER (32..65535)
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "VCI of control channel"
        ::= { ctlCktEntry 3 }

ctlCktType OBJECT-TYPE
        SYNTAX INTEGER {
               unknown(1),
               ilmi(2),
               signalling(3),
               oam(4)
               }
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Type of channel"
        ::= { ctlCktEntry 4 }

ctlCktInCells OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The Number of ATM cells received"
        ::= { ctlCktEntry 5 }

ctlCktOutCells OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The Number of ATM cells transmitted"
        ::= { ctlCktEntry 6 }

ctlCktInDiscardedClp0Cells OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The Number of ATM CLP 0 cells received and discarded"
        ::= { ctlCktEntry 7 }

ctlCktInDiscardedClp1Cells OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The Number of ATM CLP 1 cells received and discarded"
        ::= { ctlCktEntry 8 }

ctlCktInPassedClp0Cells OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The Number of ATM CLP 0 cells received and passed UPC "
        ::= { ctlCktEntry 9 }
```

```
ctlCktInPassedClp1Cells OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The Number of ATM CLP 1 cells received and passed UPC"
        ::= { ctlCktEntry 10 }

ctlCktInTaggedCells OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                 "The Number of ATM cells received and"
        ::= { ctlCktEntry 11 }

ctlCktOutClp0Cells OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The Number of ATM CLP 0 cells transmitted"
        ::= { ctlCktEntry 12 }

ctlCktOutClp1Cells OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The Number of ATM CLP 1 cells transmitted"
        ::= { ctlCktEntry 13 }
```

The cascfltsrv Group

```
--  The following Mib objects are only used by the FaultServer

fltsrvSeverity OBJECT-TYPE
        SYNTAX  INTEGER {
            critical(1),
            major(2),
            minor(3),
            warning(4),
            info (5),
            cleared(6)
              }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
            "This is the severity that the alarm is being changed to."
        ::= { cascfltsrv 1 }

fltsrvComponentID OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
            "This is the component ID for the componenet that the alarm
             transition applies to. The format of this string is :
             <switch IP>-<card number>-<Pport>-<Channel>-<Lport>-<Circuit>

             This should enable the receiver to identify the specific
             object that is effected by this alarm."
        ::= { cascfltsrv 2 }

fltsrvAlarmText OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
            "This is a text string that describes the alarm condition.
             It is assumed that the severity field will identify whether
             an alarm condition is opened or closed. An example of a
             alarm text may be Logical Port Down. A severity value of
             normal would mean that this condition had been cleared."
       ::= { cascfltsrv 3 }
```

The Traps Group

```
--  Definitions for Cascade Frame Relay Specific Traps.

nodeBoardInserted TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES  { pportSlotId, pportType }
        DESCRIPTION
                "This trap indicates that a board has been inserted into
                the indicated slot."
        ::= 1

nodeBoardPulled TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES  { pportSlotId, pportType }
        DESCRIPTION
                "This trap indicates that a board has been pulled out from
                the indicated slot."
        ::= 2

nodeBoardMismatch TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES  { pportSlotId, pportType }
        DESCRIPTION
                "This trap indicates that the actual board type on the
                indicated slot is inconsistent with what is configured."
        ::= 3
```

```
nodePsAStatusChange TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES  { nodePsAStatus }
        DESCRIPTION
                "This trap indicates that the power supply A has changed
                state (toggled between up and down states)."
        ::= 4

nodePsBStatusChange TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES  { nodePsBStatus }
        DESCRIPTION
                "This trap indicates that the power supply B has changed
                state (toggled between up and down states)."
        ::= 5

nodeFanStatusChange TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES  { nodeFanIndex, nodeFanStatus }
        DESCRIPTION
                "This trap indicates that a fan has changed state
                (toggled between up and down states)."
        ::= 6

nodeSwDownloadComplete TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { nodeSwFilename }
        DESCRIPTION
                "This trap indicates that a software download was
                complete."
        ::= 7

nodeSwDownloadFailed TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { nodeSwFilename }
        DESCRIPTION
                "This trap indicates that a software download failed."
        ::= 8

nodePrDownloadComplete TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { nodePrFilename }
        DESCRIPTION
                "This trap indicates that a PRAM download was complete."
        ::= 9

nodePrDownloadFailed TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { nodePrFilename }
        DESCRIPTION
                "This trap indicates that a PRAM download failed."
        ::= 10

nodeTracefull TRAP-TYPE
        ENTERPRISE cascfr
        DESCRIPTION
                "This trap indicates that the node trace table was full
                and has been copied to a TFTP buffer awaiting to be xfered
                to the NMS."
        ::= 11

nodeDiagLogfull TRAP-TYPE
        ENTERPRISE cascfr
        DESCRIPTION
                "This trap indicates that the diagnostic log table was
                full and has been copied to a TFTP buffer awaiting to be
                xfered to the NMS."
        ::= 12

nodeFlashMemErr TRAP-TYPE
        ENTERPRISE cascfr
        DESCRIPTION
                "This trap indicates that a checksum or CRC-32 error
                occurred in Flash."
        ::= 13
```

```
nodePramErr TRAP-TYPE
        ENTERPRISE cascfr
        DESCRIPTION
                "This trap indicates that a checksum error or battery
                problem occurred in PRAM."
        ::= 14

nodeRamErr TRAP-TYPE
        ENTERPRISE cascfr
        DESCRIPTION
                "This trap indicates that an I/O error occurred in DRAM or
                SRAM."
        ::= 15

nodeInternalErr TRAP-TYPE
        ENTERPRISE cascfr
        DESCRIPTION
                "This trap indicates that a fatal internal hardware or
                software error encountered and system needs to be re-
                booted."
        ::= 16

pportStatusChange TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { pportSlotId, pportId, pportOperStatus,
                    pportLinkDownReason }
        DESCRIPTION
                "This trap indicates that the indicated physical port has
                changed state."
        ::= 17

lportCongests TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { lportIfIndex, lportCongestRate }
        DESCRIPTION
                "This trap indicates that the rate of entering severely and
                absolutely congested state on the indicated logical port
                has exceeded the threshold."
        ::= 18

trkStatusChange TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { lportIfIndex, lportSlotId, lportPportId, lportId,
                    lportTrkStatus }
        DESCRIPTION
                "This trap indicates that the trunk associated with the
                indicated logical port has changed state."
        ::= 19

cktDlciStatusChange TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES  { cktSrcIfIndex, cktSrcDlci, cktOperStatus,
                     cktFailReason, cktFailNode, cktFailPort }
        DESCRIPTION
                "This trap indicates that the user-to-user PVC state has
                been changed for this virtual circuit. It has either been
                created or invalidated, or has toggled between the active
                and inactive states."
        ::= 20

cktDlciReroute TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES  { cktSrcIfIndex, cktSrcDlci }
        DESCRIPTION
                "This trap indicates that a PVC has been re-routed."
        ::= 21

pportInterfaceMismatch TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES  { pportSlotId, pportId, pportAdminInterface,
                     pportInterface }
        DESCRIPTION
                "This trap indicates the actual physical interface is
                different than the configured physical interface."
        ::= 22

lportErrorExceedThreshold TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES  { lportIfIndex, ifInErrors }
        DESCRIPTION
                "This trap indicates the frame errors per minute on this
                lport exceeded the threshold."
        ::= 23

nodeErrorReport TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES  { nodeDiagNonFatalSource, nodeDiagNonFatalTime,
                     nodeDiagNonFatalErrMajor, nodeDiagNonFatalErrMinor,
                     nodeDiagNonFatalStr }
        DESCRIPTION
                "A component in the switch discovered a non-fatal error
                condition."
        ::= 24

cktGrpStatusChange TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES  { cardCktGroupTrap }
        DESCRIPTION
                "Circuits on interfaces go up or down."
        ::= 25
```

```
nodeUserLogin TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES  { nodeConsoleIndex, nodeUserName, nodeUserFrom,
                     nodeConsoleUptime }
        DESCRIPTION
                "An user has logged in STDX through either serial console
                or telnet."
        ::= 26

nodeUserLogout TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES  { nodeConsoleIndex, nodeUserName, nodeUserFrom,
                     nodeConsoleUptime }
        DESCRIPTION
                "An user has logged out."
        ::= 27

cardUp TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES  { cardPhysicalSlotId }
        DESCRIPTION
                "This trap indicates that an intelligent card has
                transitioned to an ACTIVE state."
        ::= 28

cardDown TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES  { cardPhysicalSlotId }
        DESCRIPTION
                "This trap indicates that an intelligent card has
                transitioned to a NON-ACTIVE state."
        ::= 29

lnkStatusChange TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { lportIfIndex, lportSlotId, lportPportId, lportId,
                    lportLinkStatus}
        DESCRIPTION
                "This trap indicates that the link status (Link protocol)
                associated with the indicated logical port has changed
                state."
        ::= 30

lnkSmdsHbpNaTca TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { lportIfIndex, lportSlotId, lportPportId, lportId }
        DESCRIPTION
                "This trap indicates that non-Ack count for SMDS DXI
                heartbeat poll exceedes the threshold. The non-Ack count
                is reset every 15 minutes."
        ::= 31

lnkSmdsDiscardTca TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { lportIfIndex, lportSlotId, lportPportId, lportId }
        DESCRIPTION
                "This trap indicates that Total Discard packet count for
                SMDS packet exceedes the threshold. ( This trap has not
                been implemented in this release yet )."
        ::= 32

cardRedundSwitchOver TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { cardPhysicalSlotId }
        DESCRIPTION
                "This trap indicates that the card in the specified slot
                has shot its active partner and has become the active card
                of the pair."
        ::= 33

cardErrorReport TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES  { cardDiagNonFatalSource, cardDiagNonFatalTime,
                     cardDiagNonFatalErrMajor, cardDiagNonFatalErrMinor,
                     cardDiagNonFatalStr, cardPhysicalSlotId }
        DESCRIPTION
                "The specified card discovered a non-fatal error
                condition."
        ::= 34
svcSetup TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES  { cktSrcIfIndex, cktSrcDlci, cktSvcCallingParty,
                     cktSvcCalledParty }
        DESCRIPTION
                "This trap is generated by the ingress switch whenever a
                Frame Relay Svc is setup or cleared. cktStatus indicates
                whether it is a setup or a clearing."
        ::= 35

svcClearing TRAP-TYPE
        ENTERPRISE cascfr
        DESCRIPTION
                "Reserved for Frame Relay SVC."
        ::= 36

svcFailure TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES  { cktSvcCallingParty, cktSvcCalledParty, cktSvcCause }
        DESCRIPTION
                "This trap is generated by the ingress switch whenever a
                Frame Relay svc setup fails. cktSvcCause contains the
                cause number."
         ::= 37
```

```
trkBuAttempt TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { lportIfIndex, lportSlotId, lportPportId, lportId }
        DESCRIPTION
                "This trap indicates that an attempt to establish the
                backup trunk associated with the indicated logical port is
                being made."
        ::= 38

trkBuFailure TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { lportIfIndex, lportSlotId, lportPportId, lportId,
                    lportBuFailReason }
        DESCRIPTION
                "This trap indicates that the primary trunk associated
                with the indicated logical port has not been backed up, or
                the backup trunk associated with the indicated logical
                port has not been restored. The reason for failure is
                provided. The lportBuFailReason failure is provided.  The
                lportBuFailReason indicates the trunk type as follows:

                lportBuFailReason      Trunk Type
                buTrkNotDef          primary trunk
                buTrkNotEstab        backup trunk"
        ::= 39

trkBuReleased TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { lportIfIndex, lportSlotId, lportPportId, lportId }
        DESCRIPTION
                "This trap indicates that the backup trunk associated with
                the indicated logical port has been released."
        ::= 40

pportDS0LoopUpChange TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES  { pportSlotId, pportId, pportDS0LoopUpStatus }
        DESCRIPTION
                "This trap indicates one or more DS0's have gone into
                loopback.pportDS0LoopUpStatus reports the DS0's that have
                gone into loopback."
        ::= 41

pportDS0LoopDownChange TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES  { pportSlotId, pportId, pportDS0LoopDownStatus }
        DESCRIPTION
                "This trap indicates one or more DS0's have gone out of
                loopback.pportDS0LoopDownStatus reports the DS0's that have
                gone out of loopback."
                ::= 42

lportISDNCallRej TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES  { lportIfIndex, lportISDNCallRejCause }
        DESCRIPTION
                "This trap indicates that a call has been rejected due to
                either the inability to authenticate the call or lack of
                B-channel pool resources."
        ::= 43

pportdsx3LoopChange TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { pportSlotId, pportId, pportdsx3LoopStatus }
        DESCRIPTION
                "This trap indicates when the DS3 or E3 has changed its
                current loopback state."
                ::= 44

pportds1LoopChange TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { pportSlotId, pportId, pportds1LoopStatus }
        DESCRIPTION
                "This trap indicates when the DS1 has changed its current
                loopback state."
        ::= 45

pportExtClockBackup TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { pportSlotId, pportId, pportSetClkBkup }
        DESCRIPTION
                "This trap indicates that a pport has lost the external
                clock source and is switching over to the backup clock
                selected."
        ::= 46

pportExtClockRestore TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { pportSlotId, pportId }
        DESCRIPTION
                "This trap indicates that external clock is recovered and
                the pport will switch back to external clock."
        ::= 47

pportExtClkCapabilityMismatch TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { pportSlotId, pportId }
        DESCRIPTION
                "This trap indicates the port on this card (and the card
                itself) is not capable of running external clock."
        ::= 48
```

```
nodeFileTransferReport TRAP-TYPE
        ENTERPRISE cascade
        VARIABLES  { nodeFileTransferRequest, nodeFileTransferStatus }
        DESCRIPTION
                "Reporting the outcome of a previous file transfer request.
                nodeFileTransferRequest specifies the request and
                nodeFileTransferStatus gives the outcome."
        ::= 49

nodeBillingUsageRecOvflWarning TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { nodeBillingService }
        DESCRIPTION
                "Warning that a usage record counter-value overflow
                condition has occurred while aggregating usage data
                recently collected from one or more IOPs. An overflow
                condition exists when an attempt was made to update a
                usage record counter, but such an update would have
                overflowed the counter.  In this case, the usage record is
                closed and a new one is opened, if there is sufficient
                space in the service's aggregated usage data store.
                "
        ::= 50

nodeBillingUsageRecCrFailed TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { nodeBillingService }
        DESCRIPTION
                "Notification that a usage record could not be created."
        ::= 51

nodeBillingStateChange TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { nodeBillingService, nodeBilling }
        DESCRIPTION
                "Notification that billing has been enabled or disabled on
                the switch. This trap is generated at switch boot or when
                the value of the nodeBilling object is modified."
        ::= 52

lportBillingStateChange TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { nodeBillingService, lportBilling, lportIfIndex,
                    lportSlotId, lportPportId, lportId }
        DESCRIPTION
                "Notification that billing has been enabled or disabled on
                a particular logical port.  This trap is generated the
                value of the lportBilling is modified for a particular
                port;  this trap IS NOT generated when the global
                nodeBilling is modified or when at switch boot."
        ::= 53

nodeBillingSwAPCommsFailure TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { nodeBillingAPAddress }
        DESCRIPTION
                "
                Notification that switch to Adjunct Processor communication
                has failed and that a usage file transfer has failed to
                complete.
                "
        ::= 54

svcAtmFailedCall TRAP-TYPE
        ENTERPRISE cascsvc
        VARIABLES { svcAtmConfigQ93bLastCauseTx,
        svcAtmConfigQ93bLastCauseRx }
        DESCRIPTION
                "Notification that one or more SVC call failures have been
                logged in the svcAtmFailedCallTable."
        ::= 55

svcAtmSigStatusChange TRAP-TYPE
        ENTERPRISE cascsvc
        VARIABLES { svcAtmConfigIfIndex, svcAtmConfigSigOperStatus }
        DESCRIPTION
                "Notification that the status of the ATM signalling
                function
                 has changed on the port."
        ::= 56

lportCBRLineDataError TRAP-TYPE
        ENTERPRISE   cascfr
        VARIABLES    { lportIfIndex, lportSlotId, lportPportId,
                       lportId, lportStarvation, lportRecOverflow,
                       lportLossOfCellSequence, lportLossOfStructurePointer}
        DESCRIPTION
                "This trap indicates the error state of the CBR lport."
        ::= 57

clkSourceSwitch TRAP-TYPE
        ENTERPRISE    cascfr
        VARIABLES     { pportSlotId, pportId, pportCbrCurrentClockMode }
        DESCRIPTION
                "This trap indicates that the CBR port switched its clock
                method."
        ::= 58

clkSourceFailure TRAP-TYPE
        ENTERPRISE    cascfr
        VARIABLES     { pportSlotId, pportId, nodeRefclockActiveSrc }
        DESCRIPTION
                "This trap indicates that the clock generation unit
                switched between holdover mode and synchronization to
                reference clock."
        ::= 59
```

```
lportSmdsSip3SaNotFoundTca TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { lportIfIndex, lportSlotId, lportPportId, lportId }
        DESCRIPTION
                "This trap indicates that the number of Sa Not Found
                violations exceeded the specified Smds pdu violation
                threshold for thelogical port."
        ::= 60

lportSmdsSip3SaDaOnSamePortTca TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { lportIfIndex, lportSlotId, lportPportId, lportId }
        DESCRIPTION
                "This trap indicates that the number of Sa Da On Same Port
                violations exceeded the specified Smds pdu violation
                threshold for the logical port."
        ::= 61

lportSmdsSip3DstGaNotFoundTca TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { lportIfIndex, lportSlotId, lportPportId, lportId }
        DESCRIPTION
                "This trap indicates that the number of Dst Ga Not Found
                violations exceeded the specified Smds pdu violation
                threshold for the logical port."
        ::= 62

lportSmdsSip3DstIaScrnFailTca TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { lportIfIndex, lportSlotId, lportPportId, lportId }
        DESCRIPTION
                "This trap indicates that the number of Dst Ia Scrn Fail
                violations exceeded the specified Smds pdu violation
                threshold for the logical port.
                "
        ::= 63

lportSmdsSip3SaValFailTca TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { lportIfIndex, lportSlotId, lportPportId, lportId }
        DESCRIPTION
                "This trap indicates that the number of Sa Val Fail
                violations exceeded the specified Smds pdu violation
                threshold for the logical port.
                "
        ::= 64

lportSmdsSip3DstIaNotFoundTca TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { lportIfIndex, lportSlotId, lportPportId, lportId }
        DESCRIPTION
                "This trap indicates that the number of Dst Ia Not Found
                violations exceeded the specified Smds pdu violation
                threshold for the logical port.
                "
        ::= 65

lportSmdsSip3SrcIaScrnFailTca TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { lportIfIndex, lportSlotId, lportPportId, lportId }
        DESCRIPTION
                "This trap indicates that the number of Src Ia Scrn Fail
                violations exceeded the specified Smds pdu violation
                threshold for the logical port.
                "
        ::= 66

lportSmdsSip3DstGaScrnFailTca TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { lportIfIndex, lportSlotId, lportPportId, lportId }
        DESCRIPTION
                "This trap indicates that the number of Dst Ga Scrn Fail
                violations exceeded the specified Smds pdu violation
                threshold for the logical port.
                "
        ::= 67

lportSmdsDxi2LinkIdInvalidTca TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { lportIfIndex, lportSlotId, lportPportId, lportId }
        DESCRIPTION
                "This trap indicates that the number of Link Id Invalid
                violations exceeded the specified Smds pdu violation
                threshold for the logical port.
                "
        ::= 68

nodePrimarySyncReferenceChange TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { nodePrimarySyncRefOperationalState }
        DESCRIPTION
                "Notification that the Primary Synchronization Reference
                operational state has changed."
        ::= 69
```

```
nodeSecondarySyncReferenceChange TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { nodeSecondarySyncRefOperationalState }
        DESCRIPTION
                "Notification that the Secondary Synchronization Reference
                operational state has changed."
        ::= 70

nodeExternalClockAChange TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { nodeExternalClockAOperationalState }
        DESCRIPTION
                "Notification that the External Reference Clock A
                operational state has changed."
        ::= 71

nodeExternalClockBChange TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { nodeExternalClockBOperationalState }
        DESCRIPTION
                "Notification that the External Reference Clock B
                operational state has changed."
        ::= 72

nodePortReferenceAChange TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { nodePortClockAOperationalState }
        DESCRIPTION
                "Notification that the Port Reference Clock A
                operational state has changed."
        ::= 73

nodePortReferenceBChange TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { nodePortClockBOperationalState }
        DESCRIPTION
                "Notification that the Port Reference Clock B
                operational state has changed."
        ::= 74
pportDS0InitiateLpbkFailure TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { pportSlotId, pportId, pportDS0FarendDS0InLpbk }
        DESCRIPTION
                "This trap indicates that the DS0 requested to be
                set into far end loopback has failed to do so."
        ::= 75

pportDS0InitiateLpbkSuccess TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { pportSlotId, pportId, pportDS0FarendDS0InLpbk }
        DESCRIPTION
                "This trap indicates that the DS0 requested to be
                set into far end loopback has succeeded."
         ::= 76

lportISDNIpAddrRej TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES  { lportIfIndex, lportISDNIpAddrAssignFail }
        DESCRIPTION
                "This trap indicates that an Ip Address assignment failed."
        ::= 77

cktAtmStatusChange TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES  { cktSrcIfIndex, cktAtmVPI, cktAtmVCI, cktOperStatus,
                     cktFailReason, cktFailNode, cktFailPort }
        DESCRIPTION
                "This trap indicates that the user-to-user ATM PVC state
                has been changed for this virtual circuit. It has either
                been created or invalidated, or has toggled between the
                active and inactive states."
        ::= 78

cktAtmReroute TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES  { cktSrcIfIndex, cktAtmVPI, cktAtmVCI }
        DESCRIPTION
                "This trap indicates that an ATM PVC has been re-routed."
        ::= 79

cardTransmitClockStatusChange TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES  { cardPhysicalSlotId, cardTransmitClockStatus }
        DESCRIPTION
                "This trap indicates the IOM transmit clock
                synchronization status has chnaged."
        ::= 80

cardSystemPrimaryClockStatusChange TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES  { cardPhysicalSlotId, cardSystemPrimaryClockStatus }
        DESCRIPTION
                "This trap indicates that the IOM system primary clock
                reference status has changed."
        ::= 81

cardSystemSecondaryClockStatusChange TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES  { cardPhysicalSlotId, cardSystemSecondaryClockStatus }
        DESCRIPTION
                "This trap indicates that the IOM system primary clock
                reference status has changed."
        ::= 82
```

```
nodeBillingUsageDataReplnFailWarning TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { nodeBillingService }
        DESCRIPTION
                "Warning that the billing system manager on the active
                 CP+ failed to write the contents of the usage data
                 store on the redundant CP+."
        ::= 83

nodeBillingUsageDataRecoveryFailure TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { nodeBillingService }
        DESCRIPTION
                "Notification that the billing system manager on the
                 active CP+ failed to recover the contents of the usage
                 data store on boot."
        ::= 84

nodeBillingUsageDataDiscardedOnBoot TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { nodeBillingService }
        DESCRIPTION
                "Notification that the billing system manager on the
                 active CP+ discarded the usage data recovered on boot
                 because they are too old. Currently, if the data that
                 are recovered are older than 24 hours, they are
                 discarded."
        ::= 85

lportAuthenticationFailure TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { lportIfIndex, lportSlotId, lportPportId, lportId,
                                lportAuthDomainID, lportAuthFailReason }
        DESCRIPTION
                "This trap indicates a user authentication failure."
        ::= 86

lportMPBundleFailure TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { lportIfIndex, lportSlotId, lportPportId, lportId,
         lportMultilinkProtocolFailReason }
        DESCRIPTION
                "This trap indicates a PPP Multilink Protocol Bundle
                failure."
        ::= 87

lportBAPCallFailureStatus TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { lportIfIndex, lportSlotId, lportPportId, lportId,
                    lportBandwidthAllocProtocolCallFailReason }
        DESCRIPTION
                "This trap indicates a PPP Bandwidth Allocation Protocol
                 connection failure. It is the Call-Status-Code (Q.931
                cause code) sent from the calling system."
        ::= 88

lportISDNPPPNegotiationFail TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { lportIfIndex, lportSlotId, lportPportId, lportId,
                    lportISDNSourceAddr, lportISDNDestAddr,
                    lportPPPNegotiationFailCode,
                    nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
                "This trap indicates that ISDN/PPP Negotiations has
                failed."
        ::= 89

lportISDNAuthenticationFailure TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { lportIfIndex, lportSlotId, lportPportId, lportId,
                    lportISDNSourceAddr, lportISDNDestAddr,
                    lportAuthFailReason,
                    nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
                "This trap indicates a user ISDN authentication failure."
        ::= 90

lportISDNMPBundleFailure TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { lportIfIndex, lportSlotId, lportPportId, lportId,
                    lportISDNSourceAddr, lportISDNDestAddr,
                    lportMultilinkProtocolFailReason,
                    nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
                "This trap indicates a ISDN/PPP Multilink Protocol Bundle
                failure."
        ::= 91

lportISDNBAPCallFailureStatus TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { lportIfIndex, lportSlotId, lportPportId, lportId,
                lportISDNSourceAddr, lportISDNDestAddr,
                lportBandwidthAllocProtocolCallFailReason,
                 nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
                "This trap indicates an ISDN/PPP Bandwidth Allocation
                Protocol connection failure. It is the Call-Status-Code
                (Q.931 cause code) sent from the calling system."
        ::= 92
```

```
isdnCallerRejected TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { lportIfIndex, lportSlotId, isdnCallerIDAddr,
                    dvcCktGrpDialedE164Addr,
                    nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
                    "This trap represents a call from a caller
                    who is not authorized to access the circuit group."
        ::= 93

cugAddrMaxCugsExceeded TRAP-TYPE
        ENTERPRISE cascsvc
        VARIABLES { svcAddrAddress,
                    nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
                "Indicates a run-time configuration error on the
                number of CUGs allowed per address."
        ::= 94

cugPrefixMaxCugsExceeded TRAP-TYPE
        ENTERPRISE cascsvc
        VARIABLES { svcPrefixPrefix,
                    nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
                "Indicates a run-time configuration error on the
                number of CUGs allowed per prefix."
        ::= 95

cugNodePrefixMaxCugsExceeded TRAP-TYPE
        ENTERPRISE cascsvc
                VARIABLES { svcNodePrefixPrefix,
                            nodeTrapSeverity, nodeTrapSequenceNumber }
                DESCRIPTION
                    "Indicates a run-time configuration error on the
                    number of CUGs allowed per node prefix."
        ::= 96

chands1LoopChange TRAP-TYPE
        ENTERPRISE cascfr
                VARIABLES { chanSlotId, chanPortId, chanId,
                            chands1NearEndLoopConfig,
                            nodeTrapSeverity, nodeTrapSequenceNumber }
                DESCRIPTION
                    "This trap indicates when the DS1 has changed its
                    current loopback state."
        ::= 97

nodeBillingCallRecordSendFailure TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { nodeBillingService,
                    nodeTrapSeverity, nodeTrapSequenceNumber }
                DESCRIPTION
                    "Notification that switch to Accounting Server
                    communication has failed and that a usage file
                    transfer has failed to complete."
        ::= 98

lportNtmSevereCongestStatusChange TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES  { lportIfIndex, lportSevereCongestStatus,
                    nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
                "This trap indicates that there is a change of congestion
                status on a logical port on an IOM."
        ::= 99

cktNdcThreshCrossAlarmClp0 TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { cktNdcIfIndex, cktNdcSrcDlci,
                    cktNdcInDiscardClp0Cells,
                    cktNdcInDiscardClp0CellThresh,
                    nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
                "This trap is a Network Data Collection Threshold Crossing
                Alarm for the number of CLP0 cells discarded in a PVC on
                an IOM. It is generated not more than once within the 15-
                minute NDC measurement interval."
        ::= 100

cktNdcThreshCrossAlarmClp01 TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { cktNdcIfIndex, cktNdcSrcDlci,
                    cktNdcInDiscardClp01Cells,
                    cktNdcInDiscardClp01CellThresh,
                    nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
                "This trap is a Network Data Collection Threshold Crossing
                Alarm for the number of CLP0+1 cells discarded in a PVC on
                an IOM. It is generated not more than once within the 15-
                minute NDC measurement interval."
        ::= 101
```

```
nodeAuthenticationFailure TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { nodeAuthLoginUser, nodeAuthFailReason,
                    nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
                "This trap indicates a user console login authentication
                 failure."
        ::= 102

atmacctStateChange TRAP-TYPE
        ENTERPRISE cascatm
        VARIABLES { atmacctControl, nodeTrapSeverity,
                    nodeTrapSequenceNumber }
        DESCRIPTION
                "Notification that ATM accounting has been enabled or
                disabled on the switch. This trap is generated at switch
                boot or when the value of the atmacctControl object is
                modified."
        ::= 103

atmacctSwAPCommsFailure TRAP-TYPE
        ENTERPRISE cascatm
        VARIABLES { atmacctASAddressPri, atmacctASAddressSec,
                    atmacctASControl, atmacctSwASCommsFailures,
                    nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
                "Notification that switch to ATM Accounting Server
                communication has failed and that a usage file transfer has
                failed to complete."
        ::= 104

atmacctUsageRecCrFailed TRAP-TYPE
        ENTERPRISE cascatm
        VARIABLES { atmacctUsageRecCrFailures,
                    nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
                "Notification that an ATM usage record could not be
                created."
        ::= 105

atmacctLportStateChange TRAP-TYPE
        ENTERPRISE cascatm
        VARIABLES { atmacctLportSvcControl, atmacctLportIfIndex,
                      nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
                "Notification that accounting has been enabled or disabled
                on a particular logical port.  This trap is generated the
                value of the atmacctLportControl is modified for a
                particular port; this trap IS NOT generated when the
                global atmacctControl is modified or at switch boot."
        ::= 106

chands1AlarmStateChange TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { chanSlotId, chanPortId, chanId, chanLinkDownReason,
                    nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
                "This trap indicates when the DS1 has changed its current
                alarm state."
        ::= 107

pportDs1ESFDatalinkStatusChange TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { pportSlotId, pportId, pportESFDataLinkStatus,
                    nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
                "This trap indicates that the status of the DS1 ESF Data
                Link (FDL) has changed"
        ::= 108

pportPerfMonTCA TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { pportSlotId, pportId, pportPMTcaId,
                    nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
                "This trap indicates that a threshold crossing was detected
                on the performance parameter identified by the threshold
                table ID given, along with the slot and pport number"
        ::= 109

cardTypeMismatch TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES  { cardPhysicalSlotId, cardOperType,
                    nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
                "This trap indicates that the NMS view of the cardtype
                does not match the one from HW installation."
        ::= 110

nodeBulkSwAPCommsFailure TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { nodeBulkStatsCollectorAddress,
                    nodeBulkSwAPCommsFailures,
                    nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
                "Notification that switch to ATM Bulk Statistics Adjunct
                Processor communication has failed and that a data file
                transfer has failed to complete."
        ::= 111
```

```
atmacctOperStateChange TRAP-TYPE
        ENTERPRISE cascatm
        VARIABLES { atmacctControl, atmacctOperState,
                    nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
                "Notification that the operational state of the ATM
                accounting system on the switch has changed.  This trap is
                generated upon entering the non-operational state due to
                critical failure of communications to the ATM Accounting
                Server.  It is also generated upon recovery of this
                critcal condition.

                This trap is not generated upon change of the
                atmacctOperState object in conjunction with an operator
                change of the atmacctControl object."
 ::= 112

nodePsCStatusChange TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES  { nodePsCStatus, nodeTrapSeverity,
                     nodeTrapSequenceNumber }
        DESCRIPTION
                "This trap indicates that the power supply #3 has changed
                state toggled between up and down states)."
        ::= 113

nodePsMismatchTrap TRAP-TYPE
         ENTERPRISE cascfr
         VARIABLES  { nodeAdminStatus, nodeOperatingStatus,
                      nodeTrapSeverity, nodeTrapSequenceNumber }
         DESCRIPTION
                 "THIS OID IS OBSOLETE. THIS CAN BE REUSED.
                 This trap indicates that there is a mismatch in the
                number of power supplies as expected by the user and as
                actually found in the switch."
         ::= 114

atmacctASCommsStateChange TRAP-TYPE
        ENTERPRISE cascatm
        VARIABLES { atmacctASControl, atmacctASAddressPri,
                    atmacctASAddressSec, atmacctASCommsState,
                    nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
                "This trap indicates that the state of communcications to
                the ATM Accounting Server has changed."
        ::= 115

cktMultipntAtmStatusChange TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES  { cktLeafSrcIfIndex, cktLeafAtmVPI, cktLeafAtmVCI,
                     cktLeafEndpointIndex,
                     cktLeafOperStatus, cktLeafFailReason, cktLeafFailNode,
                     cktLeafFailPort, nodeTrapSeverity,
                     nodeTrapSequenceNumber }
        DESCRIPTION
                "This trap indicates that the point-to-multipoint ATM PVC
                state has been changed."
        ::= 116

nodeTimeOfDayInvalid TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
 "              Notification that the time-of-day clock on the switch is
                invalid or has not been configured.  This trap is generated
                only at CP or SP boot-time."
        ::= 117

pportAPSswitchingEvent TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { pportSlotId, pportId, pportAPSprotectionLineState,
                    nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
                "This trap indicates that a protection switching event has
                just taken place and the protection line is now in the
                indicated state."
        ::= 118

pportAPSworkingLineRestored TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { pportSlotId, pportId,
                    nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
                "This trap indicates that the specified working line
                pport has resumed carrying user traffic.  This may be
                due to an auto switch condition that has cleared or
                due to a problem detected on the protection line."
        ::= 119

pportAPSmodeMismatch TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { pportSlotId, pportId,
                    nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
                "This trap indicates that a mode mismatch has been
                detected based on this pport's APS configuration and
                the received K2 byte.  This happens when one LTE is
                configured for 1+1 APS and the other for 1:n APS.  The
                LTE configured for 1:n will fall back to 1+1 mode."
        ::= 120
```

```
pportAPSprotectionLineFailure TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { pportSlotId, pportId,
                    nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
                "This trap indicates that the protection line is now
                in a failed state.  APS switchover to protection is
                now inhibited.  If the protection line was carrying
                user traffic, it is switched back to the working line."
        ::= 121

pportAPSprotectionLineRestored TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { pportSlotId, pportId,
                    nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
                "This trap indicates that the protection line is now
                in an operational state.  APS switchover to protection
                is now possible."
        ::= 122

pportAPSprotectionByteFailure TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { pportSlotId, pportId,
                    nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
                "This trap indicates that the protection line has declared
                a protection byte failure.  This happens when a protection
                byte defect or inconsistent K1 byte is received and the
                condition persists for 2.5 seconds.  APS switchover to
                protection is inhibited.  If the protection line was
                carrying user traffic, it is switched back to the working
                line."
        ::= 123

pportAPSfarEndProtectionLineFailure TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { pportSlotId, pportId,
                    nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
                "This trap indicates that a far end protection line failure
                has been declared. This happens when the received K1 byte
                indicates SF on the protection line and the condition
                persists for 2.5 seconds."
        ::= 124

pportAPSfarEndProtectionLineCleared TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { pportSlotId, pportId,
                    nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
                "This trap indicates that the far end protection line
                failure has cleared.  This happens after 10sec. without an
                indication
                of SF on the protection line in the received K1 byte."
        ::= 125

pportAPSchannelMismatchDetected TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { pportSlotId, pportId,
                    nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
                "This trap indicates that a channel mismatch has been
                detected between the channel indicated in the received K2
                byte and the channel indicated in the transmitted K1
                byte.  Mismatches do not apply to the 1+1 unidirectional
                case.  This trap may be sent as part of the normal course
                of performing a switchover."
        ::= 126

pportAPSconfigStatusChange TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { pportSlotId, pportId, pportAPSpairedSlotId,
                    pportAPSpairedPportId, pportAPSconfigStatus,
                    nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
                "This trap indicates that the APS configuration status has
                changed.  When the pportAPSconfigStatus is indicated as
                invalid, the user should check pportAPSadminDir,
                pportAPSlineType, pportAPSrevertiveMode, and
                pportAPSwtrPeriod for a mismatched configuration between
                the two pports."
        ::= 127

nodeNtpServerError TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES {  cardNtpPeerAddr, nodeTime,
                    nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
                "This trap indicates that a time server fails to respond
                and no other time server is available. If the error occurs
                on powerup it shall be reported as critical; otherwise
                it shall be reported as a warning."
        ::= 128
```

```
nodeNtpTimeChange TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES {  nodeTime, nodeNtpOffset,
                     nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
                "This trap indicates that the time on the switch has
                changed spanning a second boundary."
        ::= 129

fltsrvAlarmTrap TRAP-TYPE
     ENTERPRISE cascfltsrv
     VARIABLES { fltsrvSeverity, fltsrvComponentID, fltsrvAlarmText }
     DESCRIPTION
         "This trap is generated by the fault server when an alarm is
         opened or closed."
     ::= 130

pportAPSdirectionModeMismatch TRAP-TYPE
        ENTERPRISE cascfr
        VARIABLES { pportSlotId, pportId,
                      nodeTrapSeverity, nodeTrapSequenceNumber }
        DESCRIPTION
                "This trap indicates that a direction mode mismatch has
                been detected on the indicated pport (the indicated pport
                is an APS protection line pport).  This happens when one
                LTE is configured for Unidirectional and the other for
                Bidirectional mode."
        ::= 131

END
```

snatlAdminEntry

```
-- This table supplies parameters for each discrete connection which
exists in the Frad.

snatlAdminTable OBJECT-TYPE
        SYNTAX SEQUENCE OF SnatlAdminEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "This table consists of configuration parameters
                for each link between an SDLC address on an Lport and
                an LLC2 connection over Frame Relay"
       ::= { snatl 1}

snatlAdminEntry OBJECT-TYPE
        SYNTAX SnatlAdminEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "A list of link parameters"
        INDEX { snatlAdminLport, snatlAdminSdlcAddress }
       ::= { snatlAdminTable 1 }

SnatlAdminEntry ::= SEQUENCE
                {
                snatlAdminLport
                    Index,
                snatlAdminSdlcAddress
                    Index,
                snatlAdminEnable
                    INTEGER,
                snatlAdminSourceMac
                    OCTET STRING,
                snatlAdminDestMac
                    OCTET STRING,
                snatlAdminSourceSap
                    INTEGER,
                snatlAdminDestSap
                    INTEGER,
                snatlAdminMaxDataSize
                    INTEGER,
                snatlAdminEncapsulationType
                    INTEGER,
                snatlAdminNodeType
                    INTEGER,
                snatlAdminActivationMode
                    INTEGER,
                snatlAdminIdBlkNum
                    OCTET STRING,
                snatlAdminFlowControlEnable
                    INTEGER,
                snatlAdminFlowControlInbound
                    INTEGER,
                natlAdminFlowControlOutbound
                    INTEGER,
                snatlAdminDlci
                    INTEGER,
                snatlAdminActivateWaitSec
                    INTEGER,
                snatlAdminActivateRetries
                    INTEGER
                }

snatlAdminLport OBJECT-TYPE
       SYNTAX Index
       ACCESS read-only
       STATUS mandatory
       DESCRIPTION
               "The logical port number on which the SDLC link resides."
       ::= { snatlAdminEntry 1 }
```

```
snatlAdminSdlcAddress OBJECT-TYPE
        SYNTAX Index
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
           "The SDLC address on this particular link"
        ::= { snatlAdminEntry 2 }

snatlAdminEnable OBJECT-TYPE
        SYNTAX INTEGER
               {
                disabled       (1),
                enabled       (2),
               deleted        (3)
               }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
               "Controls the link station sessions.  Disable and delete
               cause an orderly session disconnect sequence, where delete
               will delete this table."
       DEFVAL { disabled }
        ::= { snatlAdminEntry 3 }

snatlAdminSourceMac OBJECT-TYPE
        SYNTAX OCTET STRING (SIZE(6))
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
               "The source MAC address used with the RFC1490
                encapsulation header when bridge/BAN format is used."
        ::= { snatlAdminEntry 4 }

snatlAdminDestMac OBJECT-TYPE
        SYNTAX OCTET STRING (SIZE(6))
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
               "The destination MAC address used with the RFC1490
               encapsulation header when bridge/BAN format is used"
        ::= { snatlAdminEntry 5 }

snatlAdminSourceSap OBJECT-TYPE
        SYNTAX INTEGER (0..254)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
              "The source SAP address used with the RFC1490 encapsulation
               header "
       DEFVAL { 4 }
 ::= { snatlAdminEntry 6 }

snatlAdminDestSap  OBJECT-TYPE
        SYNTAX INTEGER (0..254)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
               "The destination SAP address used with the RFC1490
               encapsulation header"
        DEFVAL { 4 }
 ::= { snatlAdminEntry 7 }

snatlAdminMaxDataSize OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
               "The maximum information message size to be passed on
               this link. This value will initially be set to a
               configured value. The value when the link is active
               will reflect the actual value which may have been
               derived from other configuration limits, or from
               a runtime operation such as XID negotiation."
       DEFVAL { 512 }
        ::= { snatlAdminEntry 8 }

snatlAdminEncapsulationType OBJECT-TYPE
        SYNTAX INTEGER
               {
               routed         (1),
               bridged        (2)
               }
       ACCESS read-write
        STATUS mandatory
        DESCRIPTION
               "RFC1490 encapsulation type to use."
       DEFVAL { routed }
    ::= { snatlAdminEntry 9 }

snatlAdminNodeType  OBJECT-TYPE
        SYNTAX INTEGER
               {
               pu1      (1),
               pu2      (2),
                pu21     (3),
                pu4      (4)
        }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
               "A value describing the Node Type used on this link.
               Certain actions when setting up a connection require
               this parameter to be set."
       DEFVAL { pu2 }
        ::= { snatlAdminEntry 10 }
```

```
snatlAdminActivationMode OBJECT-TYPE
        SYNTAX INTEGER
                {
                remote_act   (0),
                local_sdlc (1),
                local_llc2 (2),
                local_both   (3)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This value controls the activation mode for the link.
                Default to local_sdlc. A value of 0x01 will cause the
                SDLC link to be activated locally. A value of 0x02
                will cause the LLC2 link to be activated
                locally. These values may be or'd together. If a value
                of zero is specified, it implies that the link will
                rely on remote activation"

        DEFVAL { local_sdlc }
        ::= { snatlAdminEntry 11 }

snatlAdminIdBlkNum OBJECT-TYPE
        SYNTAX OCTET STRING (SIZE(4))
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The IDBLOCK and IDNUM values used in XID negotiation, the
                first three nibbles are IDBLK, the other 5 are IDNUM
                -------------------------
                |7 | 6| 5| 4| 3| 2| 1| 0|
                -------------------------
                \IDBLK   /\IDNUM        /
                ------------------------"
        ::= { snatlAdminEntry 12 }

snatlAdminFlowControlEnable OBJECT-TYPE
        SYNTAX INTEGER
                {
                 disabled       (0),
                 enabled       (1)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                ""
        DEFVAL { enabled }
        ::= { snatlAdminEntry 13 }

snatlAdminFlowControlInbound OBJECT-TYPE
        SYNTAX INTEGER
                {
                disabled       (0),
                enabled        (1)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                ""
        DEFVAL { enabled }
        ::= { snatlAdminEntry 14 }

snatlAdminFlowControlOutbound OBJECT-TYPE
        SYNTAX INTEGER
                {
                 disabled      (0),
                enabled        (1)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                ""
        DEFVAL { enabled }
        ::= { snatlAdminEntry 15 }

snatlAdminDlci OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The DLCI on which the corresponding LLC2 link resides"
        ::= { snatlAdminEntry 16 }

snatlAdminActivateWaitSec OBJECT-TYPE
        SYNTAX INTEGER (1..65535)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                 "The amount of seconds between attempts to activate
                 the corresponding LLC2/SDLC links"
        DEFVAL { 300 }
        ::= { snatlAdminEntry 17 }

snatlAdminActivateRetries OBJECT-TYPE
        SYNTAX INTEGER (1..65535)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                 "The number of times activation of the corresponding
                 LLC2/SDLC links are attempted"
        DEFVAL { 10 }
        ::= { snatlAdminEntry 18 }
```

snatlOperEntry

```
-- This table supplies parameters for each discrete connection which
exists in the Frad.

snatlOperTable OBJECT-TYPE
        SYNTAX SEQUENCE OF SnatlOperEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "This table consists of Operational parameters
                for each link between an SDLC address on an Lport and an
                LLC2 connection over Frame Relay"
       ::= { snatl 2}

snatlOperEntry OBJECT-TYPE
        SYNTAX SnatlOperEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "A list of link parameters"
        INDEX { snatlAdminLport, snatlAdminSdlcAddress }
       ::= { snatlOperTable 1 }

SnatlOperEntry ::= SEQUENCE
                {
                snatlOperConnectionStatus
                    INTEGER,
                snatlOperCreateTime
                    TimeTicks,
                snatlOperStateChangeTime
                    TimeTicks,
                snatlOperFailCode1
                    INTEGER,
                snatlOperFailCode2
                    INTEGER,
                snatlOperRetriesLeft
                    INTEGER
                }

snatlOperConnectionStatus OBJECT-TYPE
        SYNTAX INTEGER
                {
                 inact         (1),
                 pendingactive (2),
                 active        (3),
                 pendinginact  (4)
                }
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "This parameter reflects the actual status of the
                link. The enabled state differs from active in that
                the other in-box components might be in an enabled
                state, but the end station might not be active, for
                example a remote controller may be powered off. The active
                state is achieved when the link protocol is
                in a state where information frames can be passed."
        ::= { snatlOperEntry 1 }

snatlOperCreateTime OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The time at which the mandatory link was enabled. This
                value is taken from the system clock and is implementation
                defined."
        ::= { snatlOperEntry 2 }

snatlOperStateChangeTime OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The time at which the connection status field last
                changed. The value is taken from the system clock."
        ::= { snatlOperEntry 3 }

snatlOperFailCode1 OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Indicates reason why a link failed to activate"
        ::= { snatlOperEntry 4 }

snatlOperFailCode2 OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Further failure information"
        ::= { snatlOperEntry 5 }
```

```
snatlOperRetriesLeft OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The amount of times left that activation of the SDLC/
               LLC2 linkstations will be attempted"
        ::= { snatlOperEntry 6 }
```

snasdlcPortTable

```
-- This entry augments the sdlc port configuration parameters with
-- some implementation specific items.

snasdlcPortTable OBJECT-TYPE
        SYNTAX SEQUENCE OF SnasdlcPortEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
               "This table is for parameters that were not included
               in RFC1747"
        ::= { snasdlc 1 }

snasdlcPortEntry OBJECT-TYPE
        SYNTAX SnasdlcPortEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
               "A table entry for additional SDLC parameters. This
               table is parallel to the SDLC port. They are both
               indexed on lport Interface Index"
       INDEX  { snasdlcPortLportIfIndex }
       ::= { snasdlcPortTable 1 }

SnasdlcPortEntry ::= SEQUENCE
               {
               snasdlcPortLportIfIndex
                   Index,
               snasdlcPortMaxRcvBtu
                   INTEGER,
               snasdlcPortIdleTimer
                   TimeTicks,
               snasdlcPortIdleTimerRetry
                   INTEGER,
               snasdlcPortNpRcvTimer
                   TimeTicks,
               snasdlcPortNpRcvTimerRetry
                   INTEGER,
               snasdlcPortWriteTimer
                   TimeTicks,
               snasdlcPortWriteTimerRetry
                   INTEGER,
               snasdlcPortPriFdplx
                   INTEGER,
               snasdlcPortSecFdplx
                   INTEGER,
               snasdlcPortUseRej
                   INTEGER,
               snasdlcPortMaxXidSize
                   INTEGER,
               snasdlcPortMaxRetryCount
                   INTEGER,
               snasdlcPortResetStats
                   INTEGER
               }

snasdlcPortLportIfIndex OBJECT-TYPE
        SYNTAX Index
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
               "The ifIndex value of the corresponding ifEntry."
        ::= { snasdlcPortEntry 1 }

snasdlcPortMaxRcvBtu OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
               "The absolute maximum Receive BTU per message unit allowed.
               This sort of figure is described in various other mib
               entries. The value here is intended as a cap which may
               override other configuration entries. The value should be
               initialized to the maximum value for the SDLC link.
               Dynamic alterations to this field will take effect on
               subsequently created SDLC
               ports"
        DEFVAL { 560 }
        ::= { snasdlcPortEntry 2 }

snasdlcPortIdleTimer OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
               "This timer is used as a 'watch-dog' for a completely
               inactive line.  Idle means that nothing (not even invalid
               frame data has been received.  The timer is specified in
               milliseconds."
        DEFVAL { 10000 }
        ::= { snasdlcPortEntry 3 }
```

```
snasdlcPortIdleTimerRetry OBJECT-TYPE
        SYNTAX INTEGER (1..65535)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Number of times to re-run the snasdlcPortIdleTimer before
                failure.  This is used in conjunction with
                snasdlcPortIdleTimer to provide the overall idle timeout
                period.  This is recommended to be longer than either the
                snasdlcPortNpRcvTimer or the snasdlcLsContactTimer and
                snasdlcLsDiscTimer.  A value of 65535 is used to indicate
                an unlimited retry count.  A value of 1 indicates that an
                outage should be generated after the first timer expiry. "
        DEFVAL { 10 }
        ::= { snasdlcPortEntry 4 }

snasdlcPortNpRcvTimer OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "In Secondary SDLC mode, the non-productive receive timeout
                corresponds to the time allowed for receipt of a valid
                frame from the primary.  This is usually set in
                conjunction with the retry limit to give a long timeout
                before outage (such as about 30s).  In SDLC primary mode,
                this timer is used to produce an outage when a secondary
                station produces continuous frames without setting the F-
                bit.  The timer is specified in
                milliseconds. "
        DEFVAL { 30000 }
        ::= { snasdlcPortEntry 5 }

snasdlcPortNpRcvTimerRetry OBJECT-TYPE
        SYNTAX INTEGER (1..65535)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Non-productive receive retry limit is used in conjunction
                with the timeout value to provide the overall time before
                an outage message is issued.  A value of 65535 is used to
                indicate an unlimited retry count.  A value of 1 indicates
                that an outage should be generated after the first timer
                expiry. "
        DEFVAL { 10 }
        ::= { snasdlcPortEntry 6 }

snasdlcPortWriteTimer OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The Write timeout corresponds to the maximum time that
                it takes SDLC to transmit a complete frame.  This is
                usually set in conjunction with the retry limit to give a
                long timeout before outage of about 30s.  The timer is
                specified in milliseconds. "
        DEFVAL { 30000 }
        ::= { snasdlcPortEntry 7 }

snasdlcPortWriteTimerRetry OBJECT-TYPE
        SYNTAX INTEGER (1..65535)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The Write timeout retry limit is used in conjunction with
                the timeout value to provide the overall time before an
                outage message is issued.  A value of 65535 is used to
                indicate an unlimited retry count.  A value of 1 indicates
                that an outage should be generated after the first timer
                expiry."
        DEFVAL { 5 }
        ::= { snasdlcPortEntry 8 }

snasdlcPortPriFdplx OBJECT-TYPE
        SYNTAX INTEGER
                {
                true  (1),
                false (0)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Specifies whether the primary linkstation is full duplex"
                DEFVAL { true }
        ::= { snasdlcPortEntry 9 }

snasdlcPortSecFdplx OBJECT-TYPE
        SYNTAX INTEGER
                {
                true  (1),
                false (0)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Specifies whether the secondary LS is to be full duplex"
        DEFVAL { true }
        ::= { snasdlcPortEntry 10 }
```

```
snasdlcPortUseRej OBJECT-TYPE
        SYNTAX INTEGER
                {
                allow (1),
                disallow (2)
                }
        ACCESS read-write
         STATUS mandatory
         DESCRIPTION
                "This specifies if the remote SDLC implementation can
                receive an REJ frame. This allows for differences in
                the way the remote expects to be notified of missing
                or bad frames. The effects of changing this parameter
                will happen in subsequently created SDLC ports
  (             SdlcPortAdminTable - rfc 1747)."
        DEFVAL { allow }
         ::= { snasdlcPortEntry 11 }

snasdlcPortMaxXidSize OBJECT-TYPE
        SYNTAX INTEGER (1..256)
        ACCESS read-write
         STATUS mandatory
         DESCRIPTION
                "Maximum size of an XID that will be sent or received on
                this link.  This field is present to help minimize buffer
                usage."
        DEFVAL { 256 }
         ::= { snasdlcPortEntry 12 }

snasdlcPortMaxRetryCount OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
         STATUS mandatory
         DESCRIPTION
                "Maximum number of times that a frame or group of frames
                may be retransmitted on this port before a problem is
                diagnosed."
        DEFVAL { 5 }
         ::= { snasdlcPortEntry 13 }

snasdlcPortResetStats OBJECT-TYPE
         SYNTAX INTEGER
         ACCESS read-write
         STATUS mandatory
         DESCRIPTION
                "This is a mechanism to allow the operator to reset the
                SDLC Port statistics.  Any write to this field will cause
        the statistics to be reset."
         ::= { snasdlcPortEntry 15 }
```

snasdlcLsTable

```
-- This entry augments the sdlc LS configuration parameters with
-- some implementation specific items.

snasdlcLsTable OBJECT-TYPE
        SYNTAX SEQUENCE OF SnasdlcLsEntry
         ACCESS not-accessible
         STATUS mandatory
         DESCRIPTION
           "This table is for special tuning parameters which
                are not included in RFC 1747 for SDLC. These parameters
                allow certain features to be enabled and disabled and
                certain parameters to be tuned. This is to accommodate
                certain variations which occur in implementations of
                SDLC across various vendors and products."
         ::= { snasdlc 2 }

snasdlcLsEntry OBJECT-TYPE
        SYNTAX SnasdlcLsEntry
         ACCESS not-accessible
         STATUS mandatory
         DESCRIPTION
                "A table entry for additional SDLC parameters. This
                table is parallel to the SDLC Link Station. They are both
                indexed on lport Interface Index, and SDLC Address"
       INDEX  { snasdlcLsLportIfIndex, snasdlcLsAddress }
       ::= { snasdlcLsTable 1 }
SnasdlcLsEntry ::= SEQUENCE
                {
                snasdlcLsLportIfIndex
                    Index,
                snasdlcLsAddress
                    Index,
                snasdlcLsContactTimer
                    TimeTicks,
                snasdlcLsContactTimerRetry
                    INTEGER,
                snasdlcLsContactTimer2
                    TimeTicks,
                snasdlcLsContactTimerRetry2
                    INTEGER,
                snasdlcLsDiscTimer
                    TimeTicks,
                snasdlcLsDiscTimerRetry
                    INTEGER,
                snasdlcLsNvePollTimer
                     TimeTicks,
                snasdlcLsNvePollTimerRetry
                    INTEGER,
                snasdlcLsNvePollTimer2
                    TimeTicks,
                snasdlcLsNvePollTimerRetry2
                    INTEGER,
```

```
                snasdlcLsNoRespTimer
                    TimeTicks,
                snasdlcLsNoRespTimerRetry
                    INTEGER,
                snasdlcLsRemBusyTimer
                    TimeTicks,
                snasdlcLsRemBusyTimerRetry
                    INTEGER,
                snasdlcLsRrTimer
                    TimeTicks,
                snasdlcLsPollFrame
                    INTEGER,
                snasdlcLsPollOnIFrame
                    INTEGER,
                snasdlcLsResetStats
                    INTEGER,
                snasdlcLsRole
                    INTEGER
                }

snasdlcLsLportIfIndex OBJECT-TYPE
        SYNTAX Index
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The ifIndex value of the corresponding ifEntry."
        ::= { snasdlcLsEntry 1 }

snasdlcLsAddress OBJECT-TYPE
 SYNTAX Index
 ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The address of the Link station."
        ::= { snasdlcLsEntry 2 }

snasdlcLsContactTimer OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The Contact timeout value corresponds to the timeout
                required before a SNRM or XID is retransmitted in the
                event of non-acknowledgement (used for primary SDLC
                only).  This value must be greater than the no response
                (T1) timeout value no_resp_timer described below.  The
                timer is specified in milliseconds.  This timer is also
                used for special
                pre-activation polling. "
        DEFVAL { 5000 }
        ::= { snasdlcLsEntry 3 }

snasdlcLsContactTimerRetry OBJECT-TYPE
        SYNTAX INTEGER (1..65535)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The Contact timer retry limit value corresponds to the
                number of times transmission and retransmission of a
                contact frame (such as SNRM) is allowed using the normal
                poll timer before SDLC switches to using the slow poll
                timer.  A value of 65535 is used to indicate an unlimited
                retry count.  A value of 1 indicates that the switch to
                the slow poll should be made after the first timer expiry.
                "
        DEFVAL { 5 }
        ::= { snasdlcLsEntry 4 }

snasdlcLsContactTimer2 OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Slow poll contact timer in milliseconds.  When the contact
                timer retry count expires, SDLC continues to poll using
                this timer.  This prevents leased (multi-drop) links from
                being flooded by poll frames for absent stations. "
        DEFVAL { 12000 }
        ::= { snasdlcLsEntry 5 }

snasdlcLsContactTimerRetry2 OBJECT-TYPE
        SYNTAX INTEGER (1..65535)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The slow poll contact timer retry limit value corresponds
                to the number of times transmission and retransmission of
                a contact frame (such as SNRM) is allowed on the slow
                cycle before an outage message is sent.  A value of 65535
                is used to indicate an unlimited retry count.  A value of
                1 indicates that an outage should be generated after the
                first slow poll timer expiry."
        DEFVAL { 65535 }
        ::= { snasdlcLsEntry 6 }

snasdlcLsDiscTimer OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The Disconnect timeout value corresponds to the timeout
                required before a DISC is retransmitted in the event of
                non-acknowledgement (used for primary SDLC only).  The
                timer is specified in milliseconds."
        DEFVAL { 2000 }
        ::= { snasdlcLsEntry 7 }
```

```
snasdlcLsDiscTimerRetry OBJECT-TYPE
        SYNTAX INTEGER (1..65535)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The Disconnect timer retry limit value corresponds to the
                number of times transmission and retransmission of a DISC
                is allowed.  A value of 65535 is used to indicate an
                unlimited retry count.  A value of 1 indicates that an
                outage should be generated after the first timer expiry."
       DEFVAL { 5 }
        ::= { snasdlcLsEntry 8 }



snasdlcLsNvePollTimer OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
               "The Negative poll timeout value corresponds to the timeout
                required before an adjacent secondary station (which has
                previously been removed from the polling list - because it
                has not been responding) is re-inserted into the polling
                list.  The timer is specified in milliseconds although its
                value should be set so that it runs for seconds."
       DEFVAL { 400 }
      ::= { snasdlcLsEntry 9 }

snasdlcLsNvePollTimerRetry OBJECT-TYPE
        SYNTAX INTEGER (1..65535)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The Negative poll timer retry limit value corresponds to
                the number of times a station is removed from the polling
                list on the normal poll timer before SDLC switches to
                using the slow poll timer.  A value of 65535 is used to
                indicate an unlimited retry count.  A value of 1 indicates
                that the switch to the slow poll should be made after the
                first timer expiry."
       DEFVAL { 5 }
        ::= { snasdlcLsEntry 10 }

snasdlcLsNvePollTimer2 OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The slow negative poll timer in milliseconds.  When the
                negative poll timer retry count expires, SDLC continues to
                poll using this timer.  This prevents leased (multi-drop
)               links from being flooded by poll frames for idle stations."
       DEFVAL { 1000 }
        ::= { snasdlcLsEntry 11 }

snasdlcLsNvePollTimerRetry2 OBJECT-TYPE
        SYNTAX INTEGER (1..65535)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The slow negative poll timer retry limit value
                corresponds to
                the number of times a station is removed from the polling
                list on the slow poll cycle before an outage message is
                sent.  This value is normally set to 65535 for infinite
                retry."
       DEFVAL { 65535 }
        ::= { snasdlcLsEntry 12 }
snasdlcLsNoRespTimer OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The no response (or T1) timeout value corresponds to
                the maximum time a primary station waits (after having
                sent a frame with a poll bit) for a response frame before
                trying to poll another station.  This timer is restarted
                when a frame without the F-bit is received and stopped
                only when a frame with an F-bit is received.  The timeout
                should be set to a value not
                less than twice the transmission time for the longest I-
                frame plus adjacent station frame processing time.  The
                timer is specified in milliseconds."
       DEFVAL { 30000 }
        ::= { snasdlcLsEntry 13 }

snasdlcLsNoRespTimerRetry OBJECT-TYPE
        SYNTAX INTEGER (1..65535)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
               "The no response timer retry limit value corresponds to the
                number of times an adjacent secondary station is seen to
                fail to respond before the primary sends an outage
                message.  A value of 65535 is used to indicate an
                unlimited retry count.  A value of 1 indicates that an
                outage should be generated after the first timer expiry."
       DEFVAL { 2 }
        ::= { snasdlcLsEntry 14 }
```

```
snasdlcLsRemBusyTimer OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The Remote busy timeout value corresponds to the time
                allowed for an adjacent secondary station to be in an RNR
                condition.  This is used in conjunction with the retry
                limit value to provide the overall time before an outage
                message is sent.  The timer is specified in milliseconds."
       DEFVAL { 5000 }
        ::= { snasdlcLsEntry 15 }

snasdlcLsRemBusyTimerRetry OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The Remote busy retry limit value is used in conjunction
                with the timeout value to provide the overall timeout
                before an outage message is sent.  A value of 65535 is
                used to indicate an unlimited retry count.  A value of 1
                indicates that an outage should be generated after the
                first timer expiry."
       DEFVAL { 2 }
        ::= { snasdlcLsEntry 16 }

snasdlcLsRrTimer OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The RR Turn-around timer is expressed in milliseconds. It
                is the time that SDLC waits before turning the poll bit
                around when it has no work to do.  This field is useful
                when a very fast turn around causes hardware problems on
                the link.  It can also be used to optimize link usage
                since it is often the case that the high level software
                will generate data in response to the data contained in an
                I-frame carrying the poll bit; the pause allows the data
                to be received and processed by SDLC."
       DEFVAL { 10 }
        ::= { snasdlcLsEntry 17 }

snasdlcLsPollFrame OBJECT-TYPE
        SYNTAX INTEGER
                {
                xid (191),
                disc (83),
                snrm (147),
                snrme (223),
                test (243)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The frame to use for pre-activation polling.  This is
                normally set to XID to show that polling is in the control
                of the DLC user.  However, when SDLC is primary talking to
                an old secondary implementation, it may be necessary to
                poll using some other frame.  This polling is handled by
                SDLC as part of the CONNECT_OUT processing and uses the
                frame specified here."
       DEFVAL { snrm }
        ::= { snasdlcLsEntry 18 }
snasdlcLsPollOnIFrame OBJECT-TYPE
        SYNTAX INTEGER
                {
                true  (1),
                false (0)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Flag whether this link station is permitted to send the
                poll bit on an I-frame.  This allows SDLC to work with
                certain SDLC implementations which do not handle receipt
                of I-frames carrying the poll bit."
       DEFVAL { true }
       ::= { snasdlcLsEntry 19 }

snasdlcLsResetStats OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This is a mechanism to allow the operator to reset the
                SDLC link statistics for this link. A link is defined as
                the row in the RFC 1747 table: sdlcLSStats* which is
                indexed by the same values used to index the mandatory
                field. Any write to this field will cause the statistics
                to be reset."
        ::= { snasdlcLsEntry 20 }

snasdlcLsRole OBJECT-TYPE
        SYNTAX INTEGER
                {
                primary (1),
                secondary (2),
                negotiable (3)
                }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "Specifies the initial role for all Link Stations
                activated for this port."
       DEFVAL { primary }
        ::= { snasdlcLsEntry 21 }
```

snallcPortAdminTable

```
-- This table provides connection data for the LLC2 over Frame Relay
-- connection.
-
snallcPortAdminTable OBJECT-TYPE
        SYNTAX SEQUENCE OF SnallcPortAdminEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "This table contains objects that can be
                changed to manage an LLC port.  Changing one
                 of these parameters may take effect in the
                 operating port immediately or may wait until
                 the interface is restarted depending on the
                 details of the implementation."
        ::= { snallc 1 }

snallcPortAdminEntry OBJECT-TYPE
        SYNTAX SnallcPortAdminEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
              "The record for the frame relay group of parameters"
       INDEX { snallcPortAdminLport, snallcPortAdminDlci }
       ::= { snallcPortAdminTable 1 }

SnallcPortAdminEntry ::= SEQUENCE
               {
               snallcPortAdminLport
                   Index,
               snallcPortAdminDlci
                   Index,
               snallcPortAdminMaxPDUOctets
                   INTEGER,
               snallcPortAdminMaxRetransmits
                   INTEGER,
               snallcPortAdminAckTimer
                   TimeTicks,
               snallcPortAdminPbitTimer
                   TimeTicks,
               snallcPortAdminRejTimer
                   TimeTicks,
               snallcPortAdminBusyTimer
                   TimeTicks,
               snallcPortAdminInactTimer
                   TimeTicks,
               snallcPortAdminDelayAckTimer
                   TimeTicks,
               snallcPortAdminNw
                   INTEGER,
               snallcPortAdminStatus
                   INTEGER,
               snallcPortAdminResetStats
                   INTEGER,
               snallcPortAdminQueueThreshold
                   INTEGER,
               snallcPortMaxUnackedSend
                   INTEGER,
               snallcPortMaxUnackedRecv
                   INTEGER
               }snallcPortAdminLport
                   OBJECT-TYPE
        SYNTAX Index
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
              "The logical port number associated with this port."
        ::= { snallcPortAdminEntry 1 }

snallcPortAdminDlci OBJECT-TYPE
        SYNTAX Index
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The DLCI address associated with this port"
        ::= { snallcPortAdminEntry 2 }



snallcPortAdminMaxPDUOctets OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
               "This object denotes the default maximum I PDU
               size, in octets, that LLCs on this port may
               send to and receive from their remote LLC
               partners. This count is referred to as 'N1' in
               the IEEE 802.2 specification.  This size
               includes I-Frames, UI-Frames, XIDs, and TEST
               frames.

               This port default value may be overridden by a
               non-zero value in the snallcLsAdminMaxPDUOctets object.

               The I PDU size includes all octets in a frame,
               excluding framing characters, the MAC header,
               and the LLC header."
         REFERENCE
               "ISO 8802-2 : 1989, ANSI/IEEE 802.2 - 1989
               Section 7.8.3
               Maximum Number of Octets in an I PDU, N1."
        DEFVAL { 560 }
       ::= { snallcPortAdminEntry 3 }
```

```
snallcPortAdminMaxRetransmits OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
  "             This object denotes the default value for the
                maximum number of times which LLCs on this port
                shall retry a PDU following the expiration of
                the acknowledgement timer, the P-bit timer or
                the reject timer. When these retries are
                exhausted, the link shall be declared inactive.
                This count is referred to as 'N2' in the IEEE
                802.2 specification.

                This port default value may be overridden by a
                non-zero value in the snallcLsAdminMaxRetransmits
                object."

        REFERENCE
            "ISO 8802-2 : 1989, ANSI/IEEE 802.2 - 1989
             Section 7.8.2
             Maximum Number of Transmissions, N2."
        DEFVAL { 2 }
        ::= { snallcPortAdminEntry 4 }

snallcPortAdminAckTimer OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
               "This object denotes the default value for the
               time interval during which the LLCs on this port
               shall expect to receive either:

               1) an acknowledgement to one or more outstanding I PDUs.
               2) a response PDU to an unnumbered command PDU.

               The expiration of this timer shall cause the
               unacknowledged frames to be retransmitted
               (up to N2 times).

               For Implementations that only use a single
               'T1' Value (ref. IBM Token-Ring Network
               technical reference Chapter 11).  This object
               will be used to control/read the value.

               This port default value may be overridden by a
               non-zero value in the snallcLsAdminAckTimer object."

        REFERENCE
               "ISO 8802-2 : 1989, ANSI/IEEE 802.2 - 1989
               Section 7.8.1.1
               Acknowledgment Timer"
        DEFVAL { 300 }
        ::= { snallcPortAdminEntry 5 }
snallcPortAdminPbitTimer OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
               "This object denotes the default value for the
               time interval during which the LLCs on this
               port shall expect to receive a PDU with the F
               bit set to '1' in response to a Type 2 command
               with the P bit set to '1'.

               The expiration of this timer shall cause the
               command with the poll bit to be retransmitted
               (up to N2 times).

               This port default value may be overridden by a
               non-zero value in the snallcLsAdminPbitTimer object."

        REFERENCE
               "ISO 8802-2 : 1989, ANSI/IEEE 802.2 - 1989
               Section 7.8.1.2
               P-Bit Timer"
        DEFVAL { 300 }
        ::= { snallcPortAdminEntry 6 }

snallcPortAdminRejTimer OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
               "This object denotes the default value for the
               time interval during which the LLCs on this
               port shall expect to receive a reply to a REJ
               PDU.

               The expiration of this timer shall cause the
               REJ PDU to be retransmitted (up to N2 times).

               This port default value may be overridden by a
               non-zero value in the snallcLsAdminRejTimer object."

        REFERENCE
               "ISO 8802-2 : 1989, ANSI/IEEE 802.2 - 1989
               Section 7.8.1.3
               Reject Timer"
        DEFVAL { 300 }
        ::= { snallcPortAdminEntry 7 }
```

```
snallcPortAdminBusyTimer OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object denotes the default value for the
                time interval during which the LLCs on this
                port shall expect to receive an indication that
                a busy condition at the remote LLC has
                cleared.

                The expiration of this timer causes the
                adjacent connection component to be polled.

                This port default value may be overridden by a
                non-zero value in the snallcLsAdminBusyTimer object."

        REFERENCE
                "ISO 8802-2 : 1989, ANSI/IEEE 802.2 - 1989
              Section 7.8.1.4
              Busy-State Timer"
        DEFVAL { 30000 }
        ::= { snallcPortAdminEntry 8 }

snallcPortAdminInactTimer OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object denotes the default value for the
                time interval during which the LLCs on this
                port shall expect to receive any PDU from the
                remote LLC.  This function is not described in
                the IEEE 802.2 specification.  It is listed
                in the IBM Token-Ring Network Architecture
                Reference as the Ti parameter and is widely
                implemented.

                The expiration of this timer shall cause the
                local LLC to send a PDU to the remote LLC with
                the P bit set to '1'.

                This port default value may be overridden by a
                non-zero value in the snallcLsAdminInactTimer object.

                Any value for this object less than or equal to
                the acknowledgement timer shall mean that the
                timer is not used."

        REFERENCE
                "IBM Token-Ring Network Architecture Ref. SC30-3374
                Chapter 11: Operation of Link Stations
                Inactivity Timer (Ti)"
        DEFVAL { 3000 }
        ::= { snallcPortAdminEntry 9 }

snallcPortAdminDelayAckTimer OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "This object denotes the default value for the
                time interval during which the LLCs on this
                port shall delay acknowledgment of one or more
                I PDUs (up to the value of snallcLsOperDelayAckCount).
                This function is not described in the IEEE 802.2
                specification.

                It is listed in the IBM Token-Ring Network
                Architecture Reference as the T2 parameter
                and is widely implemented.

                The expiration of this timer shall cause the
                local LLC to acknowledge all unacknowledged I
                PDUs.  A value of 0 means that an
                acknowledgement will be sent immediately.

                This object is associated with the
                snallcPortAdminDelayAckCount object and is only
                defined if that object has a value greater than
                one."

        REFERENCE
            "  IBM Token-Ring Network Architecture Ref. SC30-3374
                Chapter 11: Operation of Link Stations
                Receiver Acknowledgment Timer (T2)"
        DEFVAL { 10 }
        ::= { snallcPortAdminEntry 10 }
```

```
snallcPortAdminNw OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
             "This object denotes the default value for the
             number of IPDUs that must be acknowledged
             before the working window size (Ww) can be
             incremented by 1 when the working window is not
             equal to the maximum transmit window size (TW).
             It controls the gradual incrementing of Ww in
             congestion situations.

             This function is not described in the IEEE
             802.2 specification.  However, it is listed
             in the IBM Token-Ring Network Architecture
             Reference as the Nw parameter and is widely
             implemented."

        REFERENCE
             "IBM Token-Ring Network Architecture Ref.
             SC30-3374 Chapter 11: Operation of Link
             Stations Number of Acknowledgments Needed to
             Increment Ww (Nw)"
        DEFVAL { 4 }
        ::= { snallcPortAdminEntry 11 }

snallcPortAdminStatus  OBJECT-TYPE
       SYNTAX INTEGER
              {
              disable (1),
              enable  (2),
              delete  (3)
              }
       ACCESS read-write
        STATUS mandatory
        DESCRIPTION
              "This is a control to enable, disable or delete (0)
              the mandatory row."
        DEFVAL { disable }
        ::= { snallcPortAdminEntry 12 }

snallcPortAdminResetStats OBJECT-TYPE
       SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
              "A write to this field will cause the runtime
              counts in the statistics table to be reset."
        ::= { snallcPortAdminEntry 13 }
```

```
snallcPortAdminQueueThreshold OBJECT-TYPE
     SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
              "The Queue Threshold used by LLC on this port."
        DEFVAL { 2 }
        ::= { snallcPortAdminEntry 14 }

snallcPortMaxUnackedSend OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
              "The maximum number of unacknowledged send I-frames which
              may be outstanding for this linkstation"
        DEFVAL { 4 }
        ::= { snallcPortAdminEntry 15 }

snallcPortMaxUnackedRecv OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
              "The maximum number of unacknowledged received I-frames
              which may be outstanding for this linkstation"
        DEFVAL { 4 }
        ::= { snallcPortAdminEntry 16 }
```

snallcPortOperTable

```
-- This table provides operational data for the LLC2 over Frame Relay
-- connection.

snallcPortOperTable OBJECT-TYPE
        SYNTAX SEQUENCE OF SnallcPortOperEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
              "This table contains operational objects for LLC2 port."

nallcPortOperEntry OBJECT-TYPE
        SYNTAX SnallcPortOperEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
             "The record for the frame relay group of parameters"
       INDEX { snallcPortAdminLport, snallcPortAdminDlci }
       ::= { snallcPortOperTable 1 }
```

```
SnallcPortOperEntry ::= SEQUENCE
                  {
                  snallcPortOperStatus INTEGER
                  }

snallcPortOperStatus  OBJECT-TYPE
        SYNTAX INTEGER
               {
                inact         (1),
                pendingactive (2),
                active        (3),
                pendinginact  (4)
               }
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
               "This is an operational status."
         ::= { snallcPortOperEntry 1 }
```

snallcPortStatsTable

```
-- This table includes all the statistics in LLC2 ports.
-
snallcPortStatsTable   OBJECT-TYPE
        SYNTAX SEQUENCE OF SnallcPortStatsEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
               "Each entry in this table contains statistics
               for a specific LLC port."
        ::= { snallc 3 }

snallcPortStatsEntry   OBJECT-TYPE
        SYNTAX SnallcPortStatsEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
             "A list of statistics for an LLC port."
        INDEX { snallcPortAdminLport, snallcPortAdminDlci }
        ::= { snallcPortStatsTable 1 }

SnallcPortStatsEntry ::= SEQUENCE
               {
               snallcPortStatsTimeSecs
                   TimeTicks,
               snallcPortStatsTimeMsecs
                   TimeTicks,
               snallcPortStatsMacAddr
                   DisplayString,
               snallcPortStatsAckTimer
                   TimeTicks,
               snallcPortStatsPbitTimer
                   TimeTicks,
               snallcPortStatsT2Timer
                    TimeTicks,
               snallcPortStatsRejTimer
                   TimeTicks,
               snallcPortStatsRetryCount
                   INTEGER,
               snallcPortStatsLsCount
                   INTEGER,
               snallcPortStatsMaxUiPDUsSent
                   Counter,
               snallcPortStatsMaxUiPDUsRcvd
                   Counter
               }

    snallcPortStatsTimeSecs OBJECT-TYPE
           SYNTAX TimeTicks
           ACCESS read-only
           STATUS mandatory
           DESCRIPTION
                  "A record of the system time, in the system format,
                  at which this port became active."
           ::= { snallcPortStatsEntry 1 }
--
    snallcPortStatsTimeMsecs OBJECT-TYPE
           SYNTAX TimeTicks
           ACCESS read-only
           STATUS mandatory
           DESCRIPTION
                  "The milliseconds part of the time in ..TimeSecs"
           ::= { snallcPortStatsEntry 2 }

    snallcPortStatsMacAddr OBJECT-TYPE
           SYNTAX DisplayString
           ACCESS read-only
           STATUS mandatory
           DESCRIPTION
                  "The Mac address associated with this Port. This
                  is the source mac address used by linkstations
                  on this port."
           ::= { snallcPortStatsEntry 3 }

    snallcPortStatsAckTimer OBJECT-TYPE
           SYNTAX TimeTicks
           ACCESS read-only
           STATUS mandatory
           DESCRIPTION
                  "The value actually specified when the port was activated."
           ::= { snallcPortStatsEntry 4 }

    snallcPortStatsPbitTimer OBJECT-TYPE
           SYNTAX TimeTicks
           ACCESS read-only
           STATUS mandatory
           DESCRIPTION
                  "The actual runtime value."
           ::= { snallcPortStatsEntry 5 }
```

```
snallcPortStatsT2Timer OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Runtime value of T2."
        ::= { snallcPortStatsEntry 6 }

snallcPortStatsRejTimer OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Runtime value for rej timer."
        ::= { snallcPortStatsEntry 7 }

snallcPortStatsRetryCount OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "Runtime max_retry value."
        ::= { snallcPortStatsEntry 8 }

snallcPortStatsLsCount OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The number of linkstations curently on this port.
                This value includes linkstations which are not yet
                in an LLC2 mode ie. in XID or TEST mode."
        ::= { snallcPortStatsEntry 9 }

snallcPortStatsMaxUiPDUsSent OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "A count of LLC1 frames (Test & XID) issued on this
                port."
        ::= { snallcPortStatsEntry 10 }

snallcPortStatsMaxUiPDUsRcvd OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "A count of LLC1 frames received on this port."
        ::= { snallcPortStatsEntry 11 }
```

snallcLsAdminTable

```
-- This table includes the Link Station Administration table

snallcLsAdminTable OBJECT-TYPE
        SYNTAX SEQUENCE OF SnallcLsAdminEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
            "This table contains objects that can be
            changed to manage an LLC connection component.
            Changing one of these parameters may take
            effect in the operating link immediately or may
            wait until the link is restarted depending on
            the details of the implementation.

            Most of the objects in this read-write table
            have corresponding read-only objects in the
            snallcLsOperTable that reflect the mandatory
            operating value.

            The operating values may be different from
            these configured values if changed by XID
            negotiation or if a configured parameter was
            changed after the link was started."
        ::= { snallc 4 }

snallcLsAdminEntry OBJECT-TYPE
        SYNTAX SnallcLsAdminEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
                "A list of configured values for an LLC connection
                component."
        INDEX { snallcLsAdminLport, snallcLsAdminDlci,
                snallcLsAdminLSap, snallcLsAdminRSap,
                 snallcLsAdminLMac, snallcLsAdminRMac }
        ::= { snallcLsAdminTable 1 }
```

```
SnallcLsAdminEntry ::= SEQUENCE
                {
                snallcLsAdminLport
                    Index,
                snallcLsAdminDlci
                    Index,
                snallcLsAdminLSap
                    Index,
                snallcLsAdminRSap
                    Index,
                snallcLsAdminLMac
                    OCTET STRING,
                snallcLsAdminRMac
                    OCTET STRING,
                snallcLsAdminMaxPDUOctets
                    INTEGER,
                snallcLsAdminMaxRetransmits
                    INTEGER,
                snallcLsAdminAckTimer
                    TimeTicks,
                snallcLsAdminPbitTimer
                    TimeTicks,
                snallcLsAdminRejTimer
                    TimeTicks,
                snallcLsAdminBusyTimer
                    TimeTicks,
                snallcLsAdminInactTimer
                    TimeTicks,
                snallcLsAdminDelayAckTimer
                     TimeTicks,
                snallcLsAdminStatus
                    INTEGER,
                snallcLsMaxUnackedSend
                    INTEGER,
                snallcLsMaxUnackedRecv
                    INTEGER,
                snallcLsRole
                     INTEGER,
                snallcLsAdminTestWaitSec
                    INTEGER,
                snallcLsAdminTestRetries
                    INTEGER,
                snallcLsAdminXidWaitSec
                    INTEGER,
                snallcLsAdminXidRetries
                    INTEGER
                }
```

```
snallcLsAdminLport OBJECT-TYPE
        SYNTAX Index
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
               "The logical port number associated with this LS."
        ::= { snallcLsAdminEntry 1 }
snallcLsAdminDlci OBJECT-TYPE
        SYNTAX Index
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
              "The DLCI associated with this LS"
        ::= { snallcLsAdminEntry 2 }

snallcLsAdminLSap OBJECT-TYPE
        SYNTAX Index
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
             "This value is the address of the local SAP
             for this Connection Component."
        ::= { snallcLsAdminEntry 3 }

snallcLsAdminRSap OBJECT-TYPE
        SYNTAX Index
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
             "This value is the address of the remote SAP
             for this Connection Component."
        ::= { snallcLsAdminEntry 4 }

snallcLsAdminLMac OBJECT-TYPE
        SYNTAX OCTET STRING (SIZE (6))
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
             "This value is the local MAC address for this
             Connection Component."
        ::= { snallcLsAdminEntry 5 }

snallcLsAdminRMac OBJECT-TYPE
        SYNTAX OCTET STRING (SIZE (6))
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
             "This value is the remote MAC address for this
             Connection Component."
        ::= { snallcLsAdminEntry 6 }
```

```
snallcLsAdminMaxPDUOctets OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
             "This object denotes the maximum I PDU size, in
             octets, that this LLC SAP may send to its
             remote connection component partner. This count
             is referred to as 'N1' in the IEEE 802.2
             specification.  This size includes
             I-Frames, UI-Frames, XIDs, and TEST frames.

             The I PDU size includes all octets in a frame,
             excluding framing characters, the MAC header
             and link header."
        DEFVAL { 0 }
        ::= { snallcLsAdminEntry 7 }

snallcLsAdminMaxRetransmits OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
             "This object denotes the value for the maximum
             number of times that this LLC shall retry PDUs
             following the expiration of the acknowledgement
             timer, the P-bit timer or the reject timer.
             When these retries are exhausted, the link
             shall be declared inactive.  This count is
             referred to as 'N2' in the IEEE 802.2
             specification."
       DEFVAL { 0 }
        ::= { snallcLsAdminEntry 8 }

snallcLsAdminAckTimer OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
               "This object denotes the value for the time
               interval during which this LLC shall expect to
               receive either:

               1) an acknowledgement to one or more
               outstanding I PDUs.
               2) a response PDU to an unnumbered command PDU.

               The expiration of this timer shall cause the
               frame unacknowledged frames to be retransmitted
               (up to N2 times)."
        DEFVAL { 0 }
        ::= { snallcLsAdminEntry 9 }

snallcLsAdminPbitTimer OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
               "This object denotes the value for the time
               interval during which this LLC shall expect to
               receive a PDU with the F bit set to '1' in
               response to a Type 2 command with the P bit set
               to '1'.

               The expiration of this timer shall cause the
               REJ PDU to be retransmitted (up to N2 times)."
               DEFVAL { 0 }
        ::= { snallcLsAdminEntry 10 }

snallcLsAdminRejTimer OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
             "This object denotes the value for the time
             interval during which this LLC shall expect to
             receive a reply to a REJ PDU.

             The expiration of this timer should cause the
            REJ PDU to be retransmitted (up to N2 times)."
        DEFVAL { 0 }
        ::= { snallcLsAdminEntry 11 }

snallcLsAdminBusyTimer OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
             "This object denotes the value for the time
             interval during which this LLC shall expect to
             receive an indication that a busy condition at
             the remote LLC has cleared.

             The expiration of this timer causes the
             adjacent connection component to be polled."
        DEFVAL { 0 }
        ::= { snallcLsAdminEntry 12 }
```

```
snallcLsAdminInactTimer OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
             "This object denotes the value for the time
             interval during which this LLC shall expect to
             receive any PDU from the remote LLC.  This
             function is not described in the IEEE 802.2
             specification  but is widely implemented.

             The expiration of this timer shall cause the
             local LLC to send a PDU to the remote LLC with
             the P bit set to '1'.

             Any value for this object less than or equal to
             the acknowledgement timer shall mean that the
             timer is not used."
        DEFVAL { 0 }
        ::= { snallcLsAdminEntry 13 }

nallcLsAdminDelayAckTimer OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
             "This object denotes the value for the time
             interval during which this LLC shall delay
             acknowledgment of one or more I PDUs.  This function
             is not described in the IEEE 802.2 specification.

             It is listed in the IBM Token-Ring Network
             Architecture Reference as the T2 parameter
             and is widely implemented.

             The expiration of this timer shall cause the
             local LLC to acknowledge all unacknowledged I
             PDUs.

             This object is associated with the
             snallcLsAdminDelayAckCount object and is only
             defined if that object has a value greater than
             one."

        REFERENCE
            "IBM Token-Ring Network Architecture Ref. SC30-3374
             Chapter 11: Operation of Link Stations
             Receiver Acknowledgment Timer (T2)"
        DEFVAL { 0 }
        ::= { snallcLsAdminEntry 14 }

snallcLsAdminStatus OBJECT-TYPE
        SYNTAX INTEGER
               {
                disable (1),
               enable  (2),
               delete  (3)
               }
         ACCESS read-write
         STATUS mandatory
         DESCRIPTION
               "This object is used by a management station to
               create or delete the row entry in
               snallcLsAdminTable following the RowStatus textual
               convention.

               Upon successful creation of the row, an agent
               automatically creates a corresponding entry in
               the snallcLsAdminOperTable
               with snallcLsAdminOperState equal to 'aDM(1)'."
        DEFVAL { disable }
         ::= { snallcLsAdminEntry 15 }

snallcLsMaxUnackedSend OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
               "The maximum number of unacknowledged send I-frames which
               may be outstanding for this linkstation"
 ::= { snallcLsAdminEntry 16 }

snallcLsMaxUnackedRecv OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
               "The maximum number of unacknowledged received I-frames
               which may be outstanding for this linkstation"
        ::= { snallcLsAdminEntry 17 }

snallcLsRole OBJECT-TYPE
        SYNTAX INTEGER
               {
               primary(1),
               secondary(2),
               negotiable(3)
        }      ACCESS read-write
        STATUS mandatory
        DESCRIPTION
               "Role for linkstation"
        DEFVAL{ negotiable }
        ::= { snallcLsAdminEntry 18 }
```

```
snallcLsAdminTestWaitSec OBJECT-TYPE
        SYNTAX INTEGER (1..65535)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The number of seconds the LLC2 linkstation will wait
                between sending test frames"
        DEFVAL{ 10 }
        ::= { snallcLsAdminEntry 19 }

snallcLsAdminTestRetries OBJECT-TYPE
        SYNTAX INTEGER (1..65535)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The number of times the LLC2 linkstation will resend
                test frames before timing out"
        DEFVAL{ 100 }
        ::= { snallcLsAdminEntry 20 }

snallcLsAdminXidWaitSec OBJECT-TYPE
        SYNTAX INTEGER (1..65535)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
            "The number of seconds the LLC2 linkstation will wait
             between sending XID's"
        DEFVAL{ 10 }
        ::= { snallcLsAdminEntry 21 }

snallcLsAdminXidRetries OBJECT-TYPE
        SYNTAX INTEGER (1..65535)
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
            "The number of times the LLC2 linkstation will resend
             XID's before timing out"
        DEFVAL{ 1000 }
        ::= { snallcLsAdminEntry 22 }
```

snallcLsOperTable

```
-- This table includes the Link Station Operational table

snallcLsOperTable OBJECT-TYPE
        SYNTAX SEQUENCE OF SnallcLsOperEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
             "This table contains mandatory LLC link
             parameters.  Many of these objects have
             corresponding objects in the
             snallcLsAdminTable."
        ::= { snallc 5 }

snallcLsOperEntry OBJECT-TYPE
        SYNTAX SnallcLsOperEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
             "A list of status and control values for an
             LLC connection component."
        INDEX { snallcLsAdminLport, snallcLsAdminDlci,
                snallcLsAdminLSap, snallcLsAdminRSap,
                 snallcLsAdminRMac, snallcLsAdminLMac }
        ::= { snallcLsOperTable 1 }

SnallcLsOperEntry ::= SEQUENCE
                {
                snallcLsOperState
                    INTEGER,
                snallcLsOperMaxIPDUOctets
                    INTEGER,
                snallcLsOperCreateTime
                    TimeTicks,
                snallcLsOperLastModifyTime
                    TimeTicks,
                snallcLsOperLastFailTime
                    TimeTicks,
                snallcLsOperLastFailCause
                    INTEGER,
                snallcLsOperLastFailFRMRInfo
                    OCTET STRING,
                snallcLsOperRole
                    INTEGER
                }

snallcLsOperState OBJECT-TYPE
                SYNTAX INTEGER
                {
                aDM(1),
                setup(2),
                normal(3),
                busy(4),
                reject(5),
                await(6),
                awaitBusy(7),
                awaitReject(8),
                dConn(9),
                reset(10),
                error(11),
                conn(12),
                resetCheck(13),
                resetWait(14)
                }
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
             "This object describes the operational state of
```

```
             the LLC connection.  If the connection
             component is disabled, snallcLsOperState will be
             allowed to transition to aDM(1).  If the
             connection component is enabled, snallcLsOperState
             will be allowed to transition to normal(3)."

        REFERENCE
            "ISO 8802-2 : 1989, ANSI/IEEE 802.2 - 1989
             Section 7.9.2.1.
             Connection Component State Descriptions"
        ::= { snallcLsOperEntry 1 }

snallcLsOperMaxIPDUOctets OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
             "This object denotes the maximum I PDU size, in
              octets, that this LLC SAP may send to its
              remote connection component partner. This count
              is referred to as 'N1' in the IEEE 802.2
              specification.

              At connection setup, the remote LLC may send,
              using an XID frame, the maximum I PDU size
              which it is prepared to receive. If so, an
              implementation may choose to override the
              administered maximum PDU size with the
              dynamically learned value and should reflect
              that in this object.

              The I PDU size includes all octets in a frame,
              excluding framing characters, the MAC header
              and link header."
        ::= { snallcLsOperEntry 2 }

snallcLsOperCreateTime OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
             "This object describes the value of sysUpTime
              when this row was created."
        ::= { snallcLsOperEntry 3 }

snallcLsOperLastModifyTime OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
               "For administered connection components, this
               object describes the value of sysUpTime the
               last time this row was modified.  If the row
               has not been modified, then this value shall be
               zero.

               For dynamic connection components, this object
             identifies the time this connection component
             was created."
        ::= { snallcLsOperEntry 4 }

snallcLsOperLastFailTime OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
             "This object describes the value of sysUpTime
             the last time this connection component
             failed.  Connection component failure is
             defined as a transition to an snallcLsOperState
             value of aDM(1).  If the connection component
             has not failed, then this value shall be zero."
        ::= { snallcLsOperEntry 5 }

snallcLsOperLastFailCause OBJECT-TYPE
        SYNTAX INTEGER
               {
               undefined(1),
               rxFRMR(2),
               txFRMR(3),
               discReceived(4),
               discSent(5),
               retriesExpired(6),
               forcedShutdown(7)
               }
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
             "This enumerated object reflects the cause of
             the last failure of this LLC connection
             component.  If the connection component has not
             failed, then this object will have a value of
             undefined(1)."
        DEFVAL { undefined }
        ::= { snallcLsOperEntry 6 }

snallcLsOperLastFailFRMRInfo OBJECT-TYPE
        SYNTAX OCTET STRING (SIZE(5))
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
             "This object reflects the information field of
             the FRMR frame if the last failure for this LLC
             connection component was as a result of an
             invalid frame.  Otherwise, this field has no
             meaning."
        ::= { snallcLsOperEntry 7 }
```

```
snallcLsOperRole OBJECT-TYPE
       SYNTAX INTEGER
              {
              primary(1),
              secondary(2),
              negotiable(3)
              }
       ACCESS read-only
       STATUS mandatory
       DESCRIPTION
              "LLC Link Station Role after XID negotiation"
       ::= { snallcLsOperEntry 8 }
```

snallcLsStatsTable

```
-- This table includes the Link Station Status table                --

snallcLsStatsTable OBJECT-TYPE
      SYNTAX SEQUENCE OF SnallcLsStatsEntry
       ACCESS not-accessible
       STATUS mandatory
       DESCRIPTION
              "Each entry in this table contains statistics
              for a specific LLC connection component."
       ::= { snallc 6 }

snallcLsStatsEntry OBJECT-TYPE
       SYNTAX SnallcLsStatsEntry
       ACCESS not-accessible
       STATUS mandatory
       DESCRIPTION
            "A list of statistics for an LLC connection
            component."
       INDEX { snallcLsAdminLport, snallcLsAdminDlci,
              snallcLsAdminLSap, snallcLsAdminRSap,
               snallcLsAdminLMac, snallcLsAdminRMac }
 ::= { snallcLsStatsTable 1 }

SnallcLsStatsEntry ::= SEQUENCE
              {
              snallcLsStatsRifLen
                  INTEGER,
              snallcLsStatsRif
                  OCTET STRING,
              snallcLsStatsLsFsm
                  INTEGER,
              snallcLsStatsMacType
                  INTEGER,
              snallcLsStatsMaxPDUOctets
                  INTEGER,
              snallcLsStatsSendWindow
                  INTEGER,
              snallcLsStatsRcvWindow
                  INTEGER,
              snallcLsStatsT1Count
                  Counter,
              snallcLsStatsT2Count
                  Counter,
              snallcLsStatsRemoteBusy
                   Counter,
              snallcLsStatsIFramesOut
                  Counter,
              snallcLsStatsIOctetsOut
                  Counter,
              snallcLsStatsIFramesIn
                  Counter,
              snallcLsStatsIOctetsIn
                  Counter,
              snallcLsStatsIFramesRej
                  Counter,
              snallcLsStatsIOctetsRej
                  Counter,
              snallcLsStatsIFramesRetransmit
                  Counter,
              snallcLsStatsIOctetsRetransmit
                  Counter,
              snallcLsStatsRejFramesSent
                  Counter,
              snallcLsStatsRejFramesRcvd
                  Counter,
              snallcLsStatsXidFramesSent
                  Counter,
              snallcLsStatsXidFramesRcvd
                  Counter,
              snallcLsStatsAckTimer
                  TimeTicks,
              snallcLsStatsPbitTimer
                  TimeTicks,
              snallcLsStatsT2Timer
                  TimeTicks,
              snallcLsStatsRejTimer
                  TimeTicks,
              snallcLsStatsBusytimer
                  TimeTicks,
              snallcLsStatsInactTimer
                  TimeTicks,
              snallcLsStatsMaxRetry
                  INTEGER
              }

snallcLsStatsRifLen OBJECT-TYPE
       SYNTAX INTEGER
       ACCESS read-only
       STATUS mandatory
       DESCRIPTION
           " "
       ::= { snallcLsStatsEntry 1 }
```

```
snallcLsStatsRif OBJECT-TYPE
        SYNTAX OCTET STRING
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            " "
        ::= { snallcLsStatsEntry 2 }

snallcLsStatsLsFsm OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            " "
        ::= { snallcLsStatsEntry 3 }

snallcLsStatsMacType OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            " "
        ::= { snallcLsStatsEntry 4 }

snallcLsStatsMaxPDUOctets OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            " "
        ::= { snallcLsStatsEntry 5 }
snallcLsStatsSendWindow OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            " "
        ::= { snallcLsStatsEntry 6 }

snallcLsStatsRcvWindow OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            " "
        ::= { snallcLsStatsEntry 7 }

snallcLsStatsT1Count OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            " "
        ::= { snallcLsStatsEntry 8 }

snallcLsStatsT2Count OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            " "
        ::= { snallcLsStatsEntry 9 }

snallcLsStatsRemoteBusy OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            " "
        ::= { snallcLsStatsEntry 10 }

snallcLsStatsIFramesOut OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            " "
        ::= { snallcLsStatsEntry 11 }

snallcLsStatsIOctetsOut OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            " "
        ::= { snallcLsStatsEntry 12 }

snallcLsStatsIFramesIn OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            " "
        ::= { snallcLsStatsEntry 13 }

snallcLsStatsIOctetsIn OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            " "
        ::= { snallcLsStatsEntry 14 }
```

```
snallcLsStatsIFramesRej OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            " "
        ::= { snallcLsStatsEntry 15 }

snallcLsStatsIOctetsRej OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            " "
        ::= { snallcLsStatsEntry 16 }

snallcLsStatsIFramesRetransmit OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            " "
        ::= { snallcLsStatsEntry 17 }

snallcLsStatsIOctetsRetransmit OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            " "
        ::= { snallcLsStatsEntry 18 }

snallcLsStatsRejFramesSent OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            " "
        ::= { snallcLsStatsEntry 19 }
snallcLsStatsRejFramesRcvd OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            " "
        ::= { snallcLsStatsEntry 20 }

snallcLsStatsXidFramesSent OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            " "
        ::= { snallcLsStatsEntry 21 }

snallcLsStatsXidFramesRcvd OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            " "
        ::= { snallcLsStatsEntry 22 }

snallcLsStatsAckTimer OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            " "
        ::= { snallcLsStatsEntry 23 }

snallcLsStatsPbitTimer OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            " "
        ::= { snallcLsStatsEntry 24 }

snallcLsStatsT2Timer OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            " "
        ::= { snallcLsStatsEntry 25 }

snallcLsStatsRejTimer OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            " "
        ::= { snallcLsStatsEntry 26 }

snallcLsStatsBusytimer OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            " "
        ::= { snallcLsStatsEntry 27 }
```

```
snallcLsStatsInactTimer OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            " "
        ::= { snallcLsStatsEntry 28 }

snallcLsStatsMaxRetry OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            " "
        ::= { snallcLsStatsEntry 29 }

END
```

RFC1213-MIB DEFINITIONS ::= BEGIN

IMPORTS

```
        mgmt, NetworkAddress, IpAddress, Counter, Gauge,
        TimeTicks
                FROM RFC1155-SMI
        OBJECT-TYPE
                FROM RFC-1212;

--  This MIB module uses the extended OBJECT-TYPE macro as
--  defined in [14];
```

MIB-II (same prefix as MIB-I)

```
mib-2      OBJECT IDENTIFIER ::= { mgmt 1 }
```

Textual Conventions

```
DisplayString ::=
        OCTET STRING
-- This data type is used to model textual information taken
-- from the NVT ASCII character set.  By convention, objects
-- with this syntax are declared as having

--      SIZE (0..255)

PhysAddress ::=
        OCTET STRING
-- This data type is used to model media addresses.  For many
-- types of media, this will be in a binary representation.
-- For example, an ethernet address would be represented as
-- a string of 6 octets.
```

Groups in MIB-II

```
system          OBJECT IDENTIFIER ::= { mib-2 1 }

interfaces      OBJECT IDENTIFIER ::= { mib-2 2 }

at              OBJECT IDENTIFIER ::= { mib-2 3 }

ip              OBJECT IDENTIFIER ::= { mib-2 4 }

icmp            OBJECT IDENTIFIER ::= { mib-2 5 }

tcp             OBJECT IDENTIFIER ::= { mib-2 6 }

udp             OBJECT IDENTIFIER ::= { mib-2 7 }

egp             OBJECT IDENTIFIER ::= { mib-2 8 }
```

```
-- historical (some say hysterical)
-- cmot      OBJECT IDENTIFIER ::= { mib-2 9 }

transmission   OBJECT IDENTIFIER ::= { mib-2 10 }

snmp           OBJECT IDENTIFIER ::= { mib-2 11 }
```

The System Group

```
-- Implementation of the System group is mandatory for all
-- systems.  If an agent is not configured to have a value
-- for any of these variables, a string of length 0 is
-- returned.

sysDescr OBJECT-TYPE
        SYNTAX  DisplayString (SIZE (0..255))
        ACCESS  read-only
        STATUS  mandatory

        DESCRIPTION
                "A textual description of the entity.  This value should
                include the full name and version identification
                of the system's hardware type, software operating-system,
                and networking software. software.  It is mandatory that
                this only contain printable ASCII characters."
        ::= { system 1 }

sysObjectID OBJECT-TYPE
        SYNTAX  OBJECT IDENTIFIER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The vendor's authoritative identification of the
                network management subsystem contained in the
                entity.  This value is allocated within the SMI
                enterprises subtree (1.3.6.1.4.1) and provides an
                easy and unambiguous means for determining ‘what
                kind of box' is being managed.  For example, if
                vendor ‘Flintstones, Inc.' was assigned the
                subtree 1.3.6.1.4.1.4242, it could assign the
                identifier 1.3.6.1.4.1.4242.1.1 to its ‘Fred
                Router'."
               ::= { system 2 }

sysUpTime OBJECT-TYPE
        SYNTAX  TimeTicks
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The time (in hundredths of a second) since the
                network management portion of the system was last
                re-initialized."
        ::= { system 3 }

sysContact OBJECT-TYPE
        SYNTAX  DisplayString (SIZE (0..255))
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The textual identification of the contact person
                for this managed node, together with information
                on how to contact this person."
        ::= { system 4 }

sysName OBJECT-TYPE
        SYNTAX  DisplayString (SIZE (0..255))
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "An administratively-assigned name for this
                managed node.  By convention, this is the node's
                fully-qualified domain name."
        ::= { system 5 }

sysLocation OBJECT-TYPE
        SYNTAX  DisplayString (SIZE (0..255))
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The physical location of this node (e.g.,
                ‘telephone closet, 3rd floor')."
        ::= { system 6 }

sysServices OBJECT-TYPE
        SYNTAX  INTEGER (0..127)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A value which indicates the set of services that
                this entity primarily offers.

                The value is a sum.  This sum initially takes the
                value zero, Then, for each layer, L, in the range
                1 through 7, that this node performs transactions
                for, 2 raised to (L - 1) is added to the sum.  For
                example, a node which performs primarily routing
                functions would have a value of 4 (2^(3-1)).  In
                contrast, a node which is a host offering
                application services would have a value of 72
                (2^(4-1) + 2^(7-1)).  Note that in the context of
                the Internet suite of protocols, values should be
                calculated accordingly:

                layer           functionality
                1               physical (e.g., repeaters)
                2               datalink/subnetwork (e.g., bridges)
                3               internet (e.g., IP gateways)
                4               end-to-end  (e.g., IP hosts)
```

```
                7               applications (e.g., mail relays)

                For systems including OSI protocols, layers 5 and
                6 may also be counted."
        ::= { system 7 }
```

The Interfaces Group

```
-- Implementation of the Interfaces group is mandatory for all systems.

ifNumber OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of network interfaces (regardless of
                their current state) present on this system."
        ::= { interfaces 1 }
```

The Interfaces Table

```
-- The Interfaces table contains information on the entity's interfaces.
-- Each interface is thought of as being attached to a ‘subnetwork'. Note
-- that this term should not be confused with ‘subnet' which refers to an
-- addressing partitioning scheme used in the Internet suite of protocols.

ifTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF IfEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A list of interface entries.  The number of
                entries is given by the value of ifNumber."
               ::= { interfaces 2 }

ifEntry OBJECT-TYPE
        SYNTAX  IfEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "An interface entry containing objects at the
                subnetwork layer and below for a particular
                interface."
        INDEX   { ifIndex }
        ::= { ifTable 1 }

IfEntry ::=
        SEQUENCE {
                ifIndex
                    INTEGER,
                ifDescr
                    DisplayString,
                ifType
                    INTEGER,
                ifMtu
                    INTEGER,
                ifSpeed
                    Gauge,
                ifPhysAddress
                    PhysAddress,
                ifAdminStatus
                    INTEGER,
                ifOperStatus
                    INTEGER,
                ifLastChange
                    TimeTicks,
                ifInOctets
                    Counter,
                ifInUcastPkts
                    Counter,
                ifInNUcastPkts
                    Counter,
                ifInDiscards
                    Counter,
                ifInErrors
                    Counter,
                ifInUnknownProtos
                    Counter,
                ifOutOctets
                    Counter,
                ifOutUcastPkts
                    Counter,
                ifOutNUcastPkts
                    Counter,
                ifOutDiscards
                    Counter,
                ifOutErrors
                    Counter,
                ifOutQLen
                    Gauge,
                ifSpecific
                    OBJECT IDENTIFIER
                }

ifIndex OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A unique value for each interface.  Its value
                ranges between 1 and the value of ifNumber.  The
                value for each interface must remain constant at
                least from one re-initialization of the entity's
                network management system to the next re-
                initialization."
        ::= { ifEntry 1 }
```

```
ifDescr OBJECT-TYPE
        SYNTAX  DisplayString (SIZE (0..255))
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A textual string containing information about the
                interface.  This string should include the name of
                the manufacturer, the product name and the version
                of the hardware interface."
        ::= { ifEntry 2 }

ifType OBJECT-TYPE
        SYNTAX  INTEGER {
                other(1),               -- none of the following
                regular1822(2),
                hdh1822(3),
                ddn-x25(4),
                rfc877-x25(5),
                ethernet-csmacd(6),
                iso88023-csmacd(7),
                iso88024-tokenBus(8),
                iso88025-tokenRing(9),
                iso88026-man(10),
                starLan(11),
                proteon-10Mbit(12),
                proteon-80Mbit(13),
                hyperchannel(14),
                fddi(15),
                lapb(16),
                sdlc(17),
                ds1(18),                -- T-1
                e1(19),                 -- european equiv. of T-1
                basicISDN(20),
                primaryISDN(21),        -- proprietary serial
                propPointToPointSerial(22),
                ppp(23),
                softwareLoopback(24),
                eon(25),                -- CLNP over IP [11]
                ethernet-3Mbit(26),
                nsip(27),               -- XNS over IP
                slip(28),               -- generic SLIP
                ultra(29),              -- ULTRA technologies
                ds3(30),                -- T-3
                sip(31),                -- SMDS
                frame-relay(32)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The type of interface, distinguished according to
                the physical/link protocol(s) immediately 'below'
                the network layer in the protocol stack."
        ::= { ifEntry 3 }

ifMtu OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The size of the largest datagram which can be
                sent/received on the interface, specified in
                octets.  For interfaces that are used for
                transmitting network datagrams, this is the size
                of the largest network datagram that can be sent
                on the interface."
        ::= { ifEntry 4 }

ifSpeed OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "An estimate of the interface's current bandwidth
                in bits per second.  For interfaces which do not
                vary in bandwidth or for those where no accurate
                estimation can be made, this object should contain
                the nominal bandwidth."
        ::= { ifEntry 5 }

ifPhysAddress OBJECT-TYPE
        SYNTAX  PhysAddress
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The interface's address at the protocol layer
                immediately 'below' the network layer in the
                protocol stack.  For interfaces which do not have
                such an address (e.g., a serial line), this object
                should contain an octet string of zero length."
        ::= { ifEntry 6 }

ifAdminStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                up(1),                  -- ready to pass packets
                down(2),
                testing(3)              -- in some test mode
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The desired state of the interface.  The
                testing(3) state indicates that no operational
                packets can be passed."
        ::= { ifEntry 7 }
```

```
ifOperStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                    up(1),                  -- ready to pass packets
                    down(2),
                    testing(3)              -- in some test mode
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The current operational state of the interface.
                The testing(3) state indicates that no operational
                packets can be passed."
        ::= { ifEntry 8 }

ifLastChange OBJECT-TYPE
        SYNTAX  TimeTicks
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The value of sysUpTime at the time the interface
                entered its current operational state.  If the
                current state was entered prior to the last re-
                initialization of the local network management
                subsystem, then this object contains a zero
                value."
        ::= { ifEntry 9 }

ifInOctets OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of octets received on the
                interface, including framing characters."
        ::= { ifEntry 10 }

ifInUcastPkts OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of subnetwork-unicast packets
                delivered to a higher-layer protocol."
        ::= { ifEntry 11 }

ifInNUcastPkts OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of non-unicast (i.e., subnetwork-
                broadcast or subnetwork-multicast) packets
                delivered to a higher-layer protocol."
        ::= { ifEntry 12 }

ifInDiscards OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of inbound packets which were chosen
                to be discarded even though no errors had been
                detected to prevent their being deliverable to a
                higher-layer protocol.  One possible reason for
                discarding such a packet could be to free up
                buffer space."
        ::= { ifEntry 13 }

ifInErrors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of inbound packets that contained
                errors preventing them from being deliverable to a
                higher-layer protocol."
        ::= { ifEntry 14 }

ifInUnknownProtos OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of packets received via the interface
                which were discarded because of an unknown or
                unsupported protocol."
        ::= { ifEntry 15 }

ifOutOctets OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of octets transmitted out of the
                interface, including framing characters."
        ::= { ifEntry 16 }
```

```
ifOutUcastPkts OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of packets that higher-level
                protocols requested be transmitted to a
                subnetwork-unicast address, including those that
                were discarded or not sent."
        ::= { ifEntry 17 }

ifOutNUcastPkts OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of packets that higher-level
                protocols requested be transmitted to a non-
                unicast (i.e., a subnetwork-broadcast or
                subnetwork-multicast) address, including those
                that were discarded or not sent."
        ::= { ifEntry 18 }

ifOutDiscards OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of outbound packets which were chosen
                to be discarded even though no errors had been
                detected to prevent their being transmitted.  One
                possible reason for discarding such a packet could
                be to free up buffer space."
        ::= { ifEntry 19 }

ifOutErrors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of outbound packets that could not be
                transmitted because of errors."
        ::= { ifEntry 20 }

ifOutQLen OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The length of the output packet queue (in
                packets)."
        ::= { ifEntry 21 }

ifSpecific OBJECT-TYPE
        SYNTAX  OBJECT IDENTIFIER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A reference to MIB definitions specific to the
                particular media being used to realize the
                interface.  For example, if the interface is
                realized by an ethernet, then the value of this
                object refers to a document defining objects
                specific to ethernet.  If this information is not
                present, its value should be set to the OBJECT
                IDENTIFIER { 0 0 }, which is a syntatically valid
                object identifier, and any conformant
                implementation of ASN.1 and BER must be able to
                generate and recognize this value."
        ::= { ifEntry 22 }
```

The Address Translation Group

```
-- Implementation of the Address Translation group is
-- mandatory for all systems.  Note however that this group
-- is deprecated by MIB-II.  That is, it is being included
-- solely for compatibility with MIB-Inodes, and will most
-- likely be excluded from MIB-III nodes.  From MIB-II and
-- onwards, each network protocol group contains its own
-- address translation tables.

-- The Address Translation group contains one table which is
-- the union across all interfaces of the translation tables
-- for converting a NetworkAddress (e.g., an IP address) into
-- a subnetwork-specifi address.  For lack of a better term,
-- this document refers to such a subnetwork-specific address
-- as a ‘physical’ address.

atTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF AtEntry
        ACCESS  not-accessible
        STATUS  deprecated
        DESCRIPTION
                "The Address Translation tables contain the
                NetworkAddress to ‘physical' address equivalences.
                Some interfaces do not use translation tables for
                determining address equivalences (e.g., DDN-X.25
                has an algorithmic method); if all interfaces are
                of this type, then the Address Translation table
                is empty, i.e., has zero entries."
        ::= { at 1 }
```

```
atEntry OBJECT-TYPE
        SYNTAX  AtEntry
        ACCESS  not-accessible
        STATUS  deprecated
        DESCRIPTION
                "Each entry contains one NetworkAddress to
                'physical' address equivalence."
        INDEX   { atIfIndex,
                  atNetAddress }
        ::= { atTable 1 }

AtEntry ::=
        SEQUENCE {
                atIfIndex
                    INTEGER,
                atPhysAddress
                    PhysAddress,
                atNetAddress
                    NetworkAddress
        }

atIfIndex OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  deprecated
        DESCRIPTION
                "The interface on which this entry's equivalence
                is effective.  The interface identified by a
                particular value of this index is the same
                interface as identified by the same value of
                ifIndex."
        ::= { atEntry 1 }

atPhysAddress OBJECT-TYPE
        SYNTAX  PhysAddress
        ACCESS  read-write
        STATUS  deprecated
        DESCRIPTION
                "The media-dependent 'physical' address.

                Setting this object to a null string (one of zero
                length) has the effect of invaliding the
                corresponding entry in the atTable object.  That
                is, it effectively dissasociates the interface
                identified with said entry from the mapping
                identified with said entry.  It is an
                implementation-specific matter as to whether the
                agent removes an invalidated entry from the table.
                Accordingly, management stations must be prepared
                to receive tabular information from agents that
                corresponds to entries not currently in use.
                Proper interpretation of such entries requires
                examination of the relevant atPhysAddress object."
        ::= { atEntry 2 }

atNetAddress OBJECT-TYPE
        SYNTAX  NetworkAddress
        ACCESS  read-write
        STATUS  deprecated
        DESCRIPTION
                "The NetworkAddress (e.g., the IP address)
                corresponding to the media-dependent 'physical'
                address."
        ::= { atEntry 3 }
```

The IP Group

```
-- Implementation of the IP group is mandatory for all
-- systems.

ipForwarding OBJECT-TYPE
        SYNTAX  INTEGER {
                forwarding(1),          -- acting as a gateway
                not-forwarding(2)       -- NOT acting as a gateway
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The indication of whether this entity is acting
                as an IP gateway in respect to the forwarding of
                datagrams received by, but not addressed to, this
                entity.  IP gateways forward datagrams.  IP hosts
                do not (except those source-routed via the host).

                Note that for some managed nodes, this object may
                take on only a subset of the values possible.
                Accordingly, it is appropriate for an agent to
                return a 'badValue' response if a management
                station attempts to change this object to an
                inappropriate value."
        ::= { ip 1 }

ipDefaultTTL OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The default value inserted into the Time-To-Live
                field of the IP header of datagrams originated at
                this entity, whenever a TTL value is not supplied
                by the transport layer protocol."
        ::= { ip 2 }
```

```
ipInReceives OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of input datagrams received from
                interfaces, including those received in error."
        ::= { ip 3 }

ipInHdrErrors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of input datagrams discarded due to
                errors in their IP headers, including bad
                checksums, version number mismatch, other format
                errors, time-to-live exceeded, errors discovered
                in processing their IP options, etc."
        ::= { ip 4 }

ipInAddrErrors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of input datagrams discarded because
                the IP address in their IP header's destination
                field was not a valid address to be received at
                this entity.  This count includes invalid
                addresses (e.g., 0.0.0.0) and addresses of
                unsupported Classes (e.g., Class E).  For entities
                which are not IP Gateways and therefore do not
                forward datagrams, this counter includes datagrams
                discarded because the destination address was not
                a local address."
        ::= { ip 5 }

ipForwDatagrams OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of input datagrams for which this
                entity was not their final IP destination, as a
                result of which an attempt was made to find a
                route to forward them to that final destination.
                In entities which do not act as IP Gateways, this
                counter will include only those packets which were
                Source-Routed via this entity, and the Source-
                Route option processing was successful."
        ::= { ip 6 }

ipInUnknownProtos OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of locally-addressed datagrams
                received successfully but discarded because of an
                unknown or unsupported protocol."
        ::= { ip 7 }

ipInDiscards OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of input IP datagrams for which no
                problems were encountered to prevent their
                continued processing, but which were discarded
                (e.g., for lack of buffer space).  Note that this
                counter does not include any datagrams discarded
                while awaiting re-assembly."
        ::= { ip 8 }

ipInDelivers OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of input datagrams successfully
                delivered to IP user-protocols (including ICMP)."
        ::= { ip 9 }

ipOutRequests OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of IP datagrams which local IP
                user-protocols (including ICMP) supplied to IP in
                requests for transmission.  Note that this counter
                does not include any datagrams counted in
                ipForwDatagrams."
        ::= { ip 10 }
```

```
ipOutDiscards OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of output IP datagrams for which no
                problem was encountered to prevent their
                transmission to their destination, but which were
                discarded (e.g., for lack of buffer space).  Note
                that this counter would include datagrams counted
                in ipForwDatagrams if any such packets met this
                (discretionary) discard criterion."
        ::= { ip 11 }

ipOutNoRoutes OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of IP datagrams discarded because no route
                could be found to transmit them to their destination.
                Note that this counter includes any packets counted in
                ipForwDatagrams which meet this 'no-route' criterion.
                Note that this includes any datagarms which a host cannot
                route because all of its default gateways are down."
        ::= { ip 12 }

ipReasmTimeout OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The maximum number of seconds which received
                fragments are held while they are awaiting
                reassembly at this entity."
        ::= { ip 13 }

ipReasmReqds OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of IP fragments received which needed
                to be reassembled at this entity."
        ::= { ip 14 }

ipReasmOKs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of IP datagrams successfully re-assembled."
        ::= { ip 15 }

ipReasmFails OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of failures detected by the IP re-
                assembly algorithm (for whatever reason: timed
                out, errors, etc).  Note that this is not
                necessarily a count of discarded IP fragments
                since some algorithms (notably the algorithm in
                RFC 815) can lose track of the number of fragments
                by combining them as they are received."
        ::= { ip 16 }

ipFragOKs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of IP datagrams that have been
                successfully fragmented at this entity."
        ::= { ip 17 }

ipFragFails OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of IP datagrams that have been
                discarded because they needed to be fragmented at
                this entity but could not be, e.g., because their
                Don't Fragment flag was set."
        ::= { ip 18 }

ipFragCreates OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of IP datagram fragments that have
                been generated as a result of fragmentation at
                this entity."
        ::= { ip 19 }
```

The IP Address Table

```
-- The IP address table contains this entity's IP addressing
-- information.

ipAddrTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF IpAddrEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "The table of addressing information relevant to
                this entity's IP addresses."
        ::= { ip 20 }

ipAddrEntry OBJECT-TYPE
        SYNTAX  IpAddrEntry
        ACCESS  not-accessible
        STATUS   mandatory
        DESCRIPTION
                "The addressing information for one of this
                entity's IP addresses."
        INDEX   { ipAdEntAddr }
        ::= { ipAddrTable 1 }

IpAddrEntry ::=
        SEQUENCE {
                ipAdEntAddr
                    IpAddress,
                ipAdEntIfIndex
                    INTEGER,
                ipAdEntNetMask
                    IpAddress,
                ipAdEntBcastAddr
                    INTEGER,
                ipAdEntReasmMaxSize
                INTEGER (0..65535)
        }

ipAdEntAddr OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The IP address to which this entry's addressing
                information pertains."
        ::= { ipAddrEntry 1 }

ipAdEntIfIndex OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The index value which uniquely identifies the
                interface to which this entry is applicable.  The
                interface identified by a particular value of this
                index is the same interface as identified by the
                same value of ifIndex."
        ::= { ipAddrEntry 2 }

ipAdEntNetMask OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The subnet mask associated with the IP address of
                this entry.  The value of the mask is an IP
                address with all the network bits set to 1 and all
                the hosts bits set to 0."
        ::= { ipAddrEntry 3 }

ipAdEntBcastAddr OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The value of the least-significant bit in the IP
                broadcast address used for sending datagrams on
                the (logical) interface associated with the IP
                address of this entry.  For example, when the
                Internet standard all-ones broadcast address is
                used, the value will be 1.  This value applies to
                both the subnet and network broadcasts addresses
                used by the entity on this (logical) interface."
        ::= { ipAddrEntry 4 }

ipAdEntReasmMaxSize OBJECT-TYPE
        SYNTAX  INTEGER (0..65535)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The size of the largest IP datagram which this
                entity can re-assemble from incoming IP fragmented
                datagrams received on this interface."
::= { ipAddrEntry 5 }
```

The IP Routing Table

```
-- The IP routing table contains an entry for each route
-- presently known to this entity.

ipRouteTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF IpRouteEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "This entity's IP Routing table."
        ::= { ip 21 }

ipRouteEntry OBJECT-TYPE
        SYNTAX  IpRouteEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A route to a particular destination."
        INDEX   { ipRouteDest }
        ::= { ipRouteTable 1 }

IpRouteEntry ::=
        SEQUENCE {
                ipRouteDest
                    IpAddress,
                ipRouteIfIndex
                    INTEGER,

                ipRouteMetric1
                    INTEGER,
                ipRouteMetric2
                    INTEGER,
                ipRouteMetric3
                    INTEGER,
                ipRouteMetric4
                    INTEGER,
                ipRouteNextHop
                    IpAddress,
                ipRouteType
                    INTEGER,
                ipRouteProto
                    INTEGER,
                ipRouteAge
                    INTEGER,
                ipRouteMask
                    IpAddress,
                ipRouteMetric5
                    INTEGER,
                ipRouteInfo
                    OBJECT IDENTIFIER
        }

ipRouteDest OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The destination IP address of this route.  An
                entry with a value of 0.0.0.0 is considered a
                default route.  Multiple routes to a single
                destination can appear in the table, but access to
                such multiple entries is dependent on the table-
                access mechanisms defined by the network
                management protocol in use."
        ::= { ipRouteEntry 1 }
```

```
ipRouteIfIndex OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The index value which uniquely identifies the
                local interface through which the next hop of this
                route should be reached.  The interface identified
                by a particular value of this index is the same
                interface as identified by the same value of
                ifIndex."
        ::= { ipRouteEntry 2 }

ipRouteMetric1 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
                DESCRIPTION
                "The primary routing metric for this route.  The
                semantics of this metric are determined by the
                routing-protocol specified in the route's
                ipRouteProto value.  If this metric is not used,
                its value should be set to -1."
        ::= { ipRouteEntry 3 }

ipRouteMetric2 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "An alternate routing metric for this route.  The
                semantics of this metric are determined by the
                routing-protocol specified in the route's
                ipRouteProto value.  If this metric is not used,
                its value should be set to -1."
        ::= { ipRouteEntry 4 }

ipRouteMetric3 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "An alternate routing metric for this route.  The
                semantics of this metric are determined by the
                routing-protocol specified in the route's
                ipRouteProto value.  If this metric is not used,
                its value should be set to -1."
        ::= { ipRouteEntry 5 }

ipRouteMetric4 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "An alternate routing metric for this route.  The
                semantics of this metric are determined by the
                routing-protocol specified in the route's
                ipRouteProto value.  If this metric is not used,
                its value should be set to -1."
        ::= { ipRouteEntry 6 }

ipRouteNextHop OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The IP address of the next hop of this route.
                (In the case of a route bound to an interface
                which is realized via a broadcast media, the value
                of this field is the agent's IP address on that
                interface.)"
        ::= { ipRouteEntry 7 }

ipRouteType OBJECT-TYPE
        SYNTAX  INTEGER {
                other(1),       -- none of the following

                invalid(2),     -- an invalidated route

                direct (3),     -- route to directly
                                -- connected (sub-)network

                indirect (4)    -- route to a non-local
                                -- host/network/sub-network
                        }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The type of route.  Note that the values
                direct(3) and indirect(4) refer to the notion of
                direct and indirect routing in the IP
                architecture.

                Setting this object to the value invalid(2) has
                the effect of invalidating the corresponding entry
                in the ipRouteTable object.  That is, it
                effectively dissasociates the destination
                identified with said entry from the route
                identified with said entry.  It is an
                implementation-specific matter as to whether the
                agent removes an invalidated entry from the table.
                Accordingly, management stations must be prepared
                to receive tabular information from agents that
```

```
                corresponds to entries not currently in use.
                Proper interpretation of such entries requires
                examination of the relevant ipRouteType object."
        ::= { ipRouteEntry 8 }

ipRouteProto OBJECT-TYPE
        SYNTAX  INTEGER {
                other(1),       -- none of the following

                local (2),      -- non-protocol information, e.g.,
                                -- e.g., manually configured entries

                netmgmt (3),    -- set via a network management protocol

                icmp(4),        -- obtained via ICMP, e.g., Redirect

                                -- the remaining values are all
                                -- gateway routing protocols
                egp(5),
                ggp(6),
                hello(7),
                rip(8),
                is-is(9),
                es-is(10),
                ciscoIgrp(11),
                bbnSpfIgp(12),
                ospf(13),
                bgp(14)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The routing mechanism via which this route was
                learned.  Inclusion of values for gateway routing
                protocols is not intended to imply that hosts
                should support those protocols."
        ::= { ipRouteEntry 9 }

ipRouteAge OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The number of seconds since this route was last
                updated or otherwise determined to be correct.
                Note that no semantics of 'too old' can be implied
                except through knowledge of the routing protocol
                by which the route was learned."
                ::= { ipRouteEntry 10 }

ipRouteMask OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-write
        STATUS      mandatory
        DESCRIPTION
                "Indicate the mask to be logical-ANDed with the
                destination address before being compared to the
                value in the ipRouteDest field.  For those systems
                that do not support arbitrary subnet masks, an
                agent constructs the value of the ipRouteMask by
                determining whether the value of the correspondent
                ipRouteDest field belong to a class-A, B, or C
                network, and then using one of:

                    mask                network
                    255.0.0.0           class-A
                    255.255.0.0         class-B
                    255.255.255.0       class-C

                If the value of the ipRouteDest is 0.0.0.0 (a
                default route), then the mask value is also
                0.0.0.0.  It should be noted that all IP routing
                subsystems implicitly use this mechanism."
        ::= { ipRouteEntry 11 }

ipRouteMetric5 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "An alternate routing metric for this route.  The
                semantics of this metric are determined by the
                routing-protocol specified in the route's
                ipRouteProto value.  If this metric is not used,
                its value should be set to -1."
        ::= { ipRouteEntry 12 }
```

```
ipRouteInfo OBJECT-TYPE
        SYNTAX  OBJECT IDENTIFIER
        ACCESS  read-only
        STATUS      mandatory
        DESCRIPTION
                "A reference to MIB definitions specific to the
                particular routing protocol which is responsible
                for this route, as determined by the value
                specified in the route's ipRouteProto value.  If
                this information is not present, its value should
                be set to the OBJECT IDENTIFIER { 0 0 }, which is
                a syntatically valid object identifier, and any
                conformant implementation of ASN.1 and BER must be
                able to generate and recognize this value."
        ::= { ipRouteEntry 13 }
```

The IP Address Translation Table

```
-- The IP address translation table contain the IpAddress to
-- 'physical' address equivalences.  Some interfaces do not
-- use translation tables for determining address
-- equivalences (e.g., DDN-X.25 has an algorithmic method);
-- if all interfaces are of this type, then the Address
-- Translation table is empty, i.e., has zero entries.

ipNetToMediaTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF IpNetToMediaEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "The IP Address Translation table used for mapping
                from IP addresses to physical addresses."
        ::= { ip 22 }

ipNetToMediaEntry OBJECT-TYPE
        SYNTAX  IpNetToMediaEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "Each entry contains one IpAddress to 'physical'
                address equivalence."
        INDEX   { ipNetToMediaIfIndex,
                  ipNetToMediaNetAddress }
        ::= { ipNetToMediaTable 1 }

IpNetToMediaEntry ::=
        SEQUENCE {
                ipNetToMediaIfIndex
                    INTEGER,
                ipNetToMediaPhysAddress
                    PhysAddress,
                ipNetToMediaNetAddress
                    IpAddress,
                ipNetToMediaType
                    INTEGER
                }

ipNetToMediaIfIndex OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The interface on which this entry's equivalence
                is effective.  The interface identified by a
                particular value of this index is the same
                interface as identified by the same value of
                ifIndex."
        ::= { ipNetToMediaEntry 1 }
```

```
ipNetToMediaPhysAddress OBJECT-TYPE
        SYNTAX  PhysAddress
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The media-dependent 'physical' address."
        ::= { ipNetToMediaEntry 2 }

ipNetToMediaNetAddress OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The IpAddress corresponding to the media-
                dependent 'physical' address."
        ::= { ipNetToMediaEntry 3 }

ipNetToMediaType OBJECT-TYPE
        SYNTAX  INTEGER {
                other(1),       -- none of the following
                invalid(2),     -- an invalidated mapping
                dynamic(3),
                static(4)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The type of mapping.  Setting this object to the value
                invalid(2) has the effect of invalidating the corres-
                ponding entry in the ipNetToMediaTable.  That is, it
                effectively disassociates the interface identified with
                said entry from the mapping identified with said entry.
                It is an implementation-specific matter as to whether the
                agent removes an invalidated entry from the table.
                Accordingly, management stations must be prepared to
                receive tabular information from agents that
                corresponds to entries not currently in use.  Proper
                interpretation of such entries requires examination
                of the relevant ipNetToMediaType object."
        ::= { ipNetToMediaEntry 4 }
```

Additional IP Objects

```
ipRoutingDiscards OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of routing entries which were chosen
                to be discarded even though they are valid.  One
                possible reason for discarding such an entry could
                be to free-up buffer space for other routing
                entries."
        := { ip 23 }
```

The ICMP Group

```
 -- Implementation of the ICMP group is mandatory for all
-- systems.

icmpInMsgs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of ICMP messages which the
                entity received.  Note that this counter includes
                all those counted by icmpInErrors."
        ::= { icmp 1 }

icmpInErrors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of ICMP messages which the entity
                received but determined as having ICMP-specific
                errors (bad ICMP checksums, bad length, etc.)."
        ::= { icmp 2 }

icmpInDestUnreachs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of ICMP Destination Unreachable
                messages received."
        ::= { icmp 3 }

icmpInTimeExcds OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of ICMP Time Exceeded messages
                received."
        ::= { icmp 4 }

icmpInParmProbs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of ICMP Parameter Problem messages
                received."
        ::= { icmp 5 }
```

```
icmpInSrcQuenchs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
        "The number of ICMP Source Quench messages
        received."
        ::= { icmp 6 }

icmpInRedirects OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of ICMP Redirect messages received."
        ::= { icmp 7 }

icmpInEchos OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of ICMP Echo (request) messages
                received."
              ::= { icmp 8 }

icmpInEchoReps OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of ICMP Echo Reply messages received."
        ::= { icmp 9 }

icmpInTimestamps OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of ICMP Timestamp (request) messages
                received."
        ::= { icmp 10 }

icmpInTimestampReps OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of ICMP Timestamp Reply messages
                received."
        ::= { icmp 11 }

icmpInAddrMasks OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of ICMP Address Mask Request messages
                received."
        ::= { icmp 12 }

icmpInAddrMaskReps OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of ICMP Address Mask Reply messages
                received."
        ::= { icmp 13 }

icmpOutMsgs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of ICMP messages which this
                entity attempted to send.  Note that this counter
                includes all those counted by icmpOutErrors."
        ::= { icmp 14 }

icmpOutErrors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of ICMP messages which this entity did
                not send due to problems discovered within ICMP
                such as a lack of buffers.  This value should not
                include errors discovered outside the ICMP layer
                such as the inability of IP to route the resultant
                datagram.  In some implementations there may be no
                types of error which contribute to this counter's
                value."
        ::= { icmp 15 }

icmpOutDestUnreachs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of ICMP Destination Unreachable
                messages sent."
        ::= { icmp 16 }
```

```
icmpOutTimeExcds OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of ICMP Time Exceeded messages sent."
        ::= { icmp 17 }

icmpOutParmProbs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of ICMP Parameter Problem messages
                sent."
        ::= { icmp 18 }

icmpOutSrcQuenchs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of ICMP Source Quench messages sent."
        ::= { icmp 19 }

icmpOutRedirects OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of ICMP Redirect messages sent.  For a
                host, this object will always be zero, since hosts
                do not send redirects."
        ::= { icmp 20 }

icmpOutEchos OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of ICMP Echo (request) messages sent."
        ::= { icmp 21 }

icmpOutEchoReps OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
                DESCRIPTION
                "The number of ICMP Echo Reply messages sent."
        ::= { icmp 22 }

icmpOutTimestamps OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of ICMP Timestamp (request) messages sent."
        ::= { icmp 23 }

icmpOutTimestampReps OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of ICMP Timestamp Reply messages sent."
        ::= { icmp 24 }

icmpOutAddrMasks OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of ICMP Address Mask Request messages sent."
        ::= { icmp 25 }

icmpOutAddrMaskReps OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of ICMP Address Mask Reply messages sent."
        ::= { icmp 26 }
```

The TCP Group

```
-- Implementation of the TCP group is mandatory for all
-- systems that implement the TCP.

-- Note that instances of object types that represent information
-- about a particular TCP connection are transient; they persist
-- only as long as the connection in question.

tcpRtoAlgorithm OBJECT-TYPE
        SYNTAX  INTEGER {
                other(1),       -- none of the following
                constant(2),    -- a constant rto
                rsre(3),        -- MIL-STD-1778, Appendix B
                vanj(4)         -- Van Jacobson's algorithm [10]
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The algorithm used to determine the timeout value
                used for retransmitting unacknowledged octets."
        ::= { tcp 1 }
```

```
tcpRtoMin OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The minimum value permitted by a TCP
                implementation for the retransmission timeout,
                measured in milliseconds.  More refined semantics
                for objects of this type depend upon the algorithm
                used to determine the retransmission timeout.  In
                particular, when the timeout algorithm is rsre(3),
                an object of this type has the semantics of the
                LBOUND quantity described in RFC 793."
        ::= { tcp 2 }

tcpRtoMax OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The maximum value permitted by a TCP
                implementation for the retransmission timeout,
                measured in milliseconds.  More refined semantics
                for objects of this type depend upon the algorithm
                used to determine the retransmission timeout.  In
                particular, when the timeout algorithm is rsre(3),
                an object of this type has the semantics of the
                UBOUND quantity described in RFC 793."
        ::= { tcp 3 }

tcpMaxConn OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The limit on the total number of TCP connections
                the entity can support.  In entities where the
                maximum number of connections is dynamic, this
        object should contain the value -1."
        ::= { tcp 4 }

tcpActiveOpens OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of times TCP connections have made a
                direct transition to the SYN-SENT state from the
                CLOSED state."
        ::= { tcp 5 }

tcpPassiveOpens OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of times TCP connections have made a
                direct transition to the SYN-RCVD state from the
                LISTEN state."
        ::= { tcp 6 }

tcpAttemptFails OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of times TCP connections have made a
                direct transition to the CLOSED state from either
                the SYN-SENT state or the SYN-RCVD state, plus the
                number of times TCP connections have made a direct
                transition to the LISTEN state from the SYN-RCVD
                state."
        ::= { tcp 7 }

tcpEstabResets OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of times TCP connections have made a
                direct transition to the CLOSED state from either
                the ESTABLISHED state or the CLOSE-WAIT state."
        ::= { tcp 8 }

tcpCurrEstab OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of TCP connections for which the
                current state is either ESTABLISHED or CLOSE-
                WAIT."
        ::= { tcp 9 }

tcpInSegs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of segments received, including
                those received in error.  This count includes
                segments received on currently established
                connections."
        ::= { tcp 10 }
```

```
tcpOutSegs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of segments sent, including
                those on current connections but excluding those
                containing only retransmitted octets."
        ::= { tcp 11 }

tcpRetransSegs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of segments retransmitted - that
                is, the number of TCP segments transmitted
                containing one or more previously transmitted
                octets."
        ::= { tcp 12 }
```

The TCP Connection Table

```
-- The TCP connection table contains information about this
-- entity's existing TCP connections.

tcpConnTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF TcpConnEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A table containing TCP connection-specific
                information."
        ::= { tcp 13 }

tcpConnEntry OBJECT-TYPE
        SYNTAX  TcpConnEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "Information about a particular current TCP
                connection.  An object of this type is transient,
                in that it ceases to exist when (or soon after)
                the connection makes the transition to the CLOSED
                state."
        INDEX   { tcpConnLocalAddress,
                  tcpConnLocalPort,
                  tcpConnRemAddress,
                  tcpConnRemPort }
        ::= { tcpConnTable 1 }

TcpConnEntry ::=
        SEQUENCE {
                tcpConnState
                    INTEGER,
                tcpConnLocalAddress
                    IpAddress,
                tcpConnLocalPort
                    INTEGER (0..65535),
                tcpConnRemAddress
                    IpAddress,
                tcpConnRemPort
                    INTEGER (0..65535)
                }

tcpConnState OBJECT-TYPE
        SYNTAX  INTEGER {
                closed(1),
                listen(2),
                synSent(3),
                synReceived(4),
                established(5),
                finWait1(6),
                finWait2(7),
                closeWait(8),
                lastAck(9),
                closing(10),
                timeWait(11),
                deleteTCB(12)
                }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The state of this TCP connection.

                The only value which may be set by a management
                station is deleteTCB(12).  Accordingly, it is
                appropriate for an agent to return a 'badValue'
                response if a management station attempts to set
                this object to any other value.

                If a management station sets this object to the
                value deleteTCB(12), then this has the effect of
                deleting the TCB (as defined in RFC 793) of the
                corresponding connection on the managed node,
                resulting in immediate termination of the
                connection.

                As an implementation-specific option, a RST
                segment may be sent from the managed node to the
                other TCP endpoint (note however that RST segments
                are not sent reliably)."
        ::= { tcpConnEntry 1 }
```

```
tcpConnLocalAddress OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The local IP address for this TCP connection.  In
                the case of a connection in the listen state which
                is willing to accept connections for any IP
                interface associated with the node, the value
                0.0.0.0 is used."
        ::= { tcpConnEntry 2 }

tcpConnLocalPort OBJECT-TYPE
        SYNTAX  INTEGER (0..65535)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The local port number for this TCP connection."
        ::= { tcpConnEntry 3 }

tcpConnRemAddress OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The remote IP address for this TCP connection."
        ::= { tcpConnEntry 4 }

tcpConnRemPort OBJECT-TYPE
        SYNTAX  INTEGER (0..65535)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The remote port number for this TCP connection."
        ::= { tcpConnEntry 5 }
```

Additional TCP Objects

```
tcpInErrs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of segments received in error
                (e.g., bad TCP checksums)."
        ::= { tcp 14 }

tcpOutRsts OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of TCP segments sent containing the RST flag."
        ::= { tcp 15 }
```

The UDP Group

```
-- Implementation of the UDP group is mandatory for all
-- systems which implement the UDP.

udpInDatagrams OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of UDP datagrams delivered to UDP users."
        ::= { udp 1 }

udpNoPorts OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of received UDP datagrams for which
                there was no application at the destination port."
        ::= { udp 2 }

udpInErrors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of received UDP datagrams that could
                not be delivered for reasons other than the lack
                of an application at the destination port."
        ::= { udp 3 }

udpOutDatagrams OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of UDP datagrams sent from this
                entity."
        ::= { udp 4 }
```

The UDP Listener Table

```
-- The UDP listener table contains information about this
-- entity's UDP end-points on which a local application is
-- currently accepting datagrams.

udpTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF UdpEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A table containing UDP listener information."
        ::= { udp 5 }

udpEntry OBJECT-TYPE
        SYNTAX  UdpEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "Information about a particular current UDP
                listener."
        INDEX   { udpLocalAddress, udpLocalPort }
        ::= { udpTable 1 }

UdpEntry ::=
        SEQUENCE {
                udpLocalAddress
                    IpAddress,
                udpLocalPort
                    INTEGER (0..65535)
        }

udpLocalAddress OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The local IP address for this UDP listener.  In
                the case of a UDP listener which is willing to
                accept datagrams for any IP interface associated
                with the node, the value 0.0.0.0 is used."
        ::= { udpEntry 1 }

udpLocalPort OBJECT-TYPE
        SYNTAX  INTEGER (0..65535)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The local port number for this UDP listener."
        ::= { udpEntry 2 }
```

The EGP Group

```
-- Implementation of the EGP group is mandatory for all
-- systems which implement the EGP.

egpInMsgs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of EGP messages received without error."
        ::= { egp 1 }

egpInErrors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of EGP messages received that proved
                to be in error."
        ::= { egp 2 }

egpOutMsgs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of locally generated EGP messages."
        ::= { egp 3 }

egpOutErrors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS      mandatory
        DESCRIPTION
                "The number of locally generated EGP messages not
                sent due to resource limitations within an EGP
                entity."
        ::= { egp 4 }
```

The EGP Neighbor Table

```
-- The EGP neighbor table contains information about this
-- entity's EGP neighbors.

egpNeighTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF EgpNeighEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "The EGP neighbor table."
        ::= { egp 5 }


egpNeighEntry OBJECT-TYPE
```

```
        SYNTAX  EgpNeighEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "Information about this entity's relationship with
                a particular EGP neighbor."
        INDEX   { egpNeighAddr }
        ::= { egpNeighTable 1 }

EgpNeighEntry ::=
        SEQUENCE {
                egpNeighState
                    INTEGER,
                egpNeighAddr
                    IpAddress,
                egpNeighAs
                    INTEGER,
                egpNeighInMsgs
                    Counter,
                egpNeighInErrs
                    Counter,
                egpNeighOutMsgs
                    Counter,
                egpNeighOutErrs
                    Counter,
                egpNeighInErrMsgs
                    Counter,
                egpNeighOutErrMsgs
                    Counter,
                egpNeighStateUps
                    Counter,
                egpNeighStateDowns
                    Counter,
                egpNeighIntervalHello
                    INTEGER,
                egpNeighIntervalPoll
                    INTEGER,
                egpNeighMode
                    INTEGER,
                egpNeighEventTrigger
                    INTEGER
        }
egpNeighState OBJECT-TYPE
        SYNTAX  INTEGER {
                idle(1),
                acquisition(2),
                down(3),
                up(4),
                cease(5)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The EGP state of the local system with respect to
                this entry's EGP neighbor.  Each EGP state is
                represented by a value that is one greater than
                the numerical value associated with said state in
                RFC 904."
        ::= { egpNeighEntry 1 }

egpNeighAddr OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The IP address of this entry's EGP neighbor."
        ::= { egpNeighEntry 2 }

egpNeighAs OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The autonomous system of this EGP peer.  Zero
                should be specified if the autonomous system
                number of the neighbor is not yet known."
        ::= { egpNeighEntry 3 }

egpNeighInMsgs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of EGP messages received without error
                from this EGP peer."
        ::= { egpNeighEntry 4 }
egpNeighInErrs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of EGP messages received from this EGP
                peer that proved to be in error (e.g., bad EGP checksum)."
               ::= { egpNeighEntry 5 }

egpNeighOutMsgs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of locally generated EGP messages to
                this EGP peer."
        ::= { egpNeighEntry 6 }
egpNeighOutErrs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
```

```
                "The number of locally generated EGP messages not
                sent to this EGP peer due to resource limitations
                within an EGP entity."
        ::= { egpNeighEntry 7 }

egpNeighInErrMsgs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of EGP-defined error messages received
                from this EGP peer."
        ::= { egpNeighEntry 8 }

egpNeighOutErrMsgs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of EGP-defined error messages sent to
                this EGP peer."
        ::= { egpNeighEntry 9 }

egpNeighStateUps OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of EGP state transitions to the UP
                state with this EGP peer."
        ::= { egpNeighEntry 10 }
egpNeighStateDowns OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of EGP state transitions from the UP
                state to any other state with this EGP peer."
        ::= { egpNeighEntry 11 }

egpNeighIntervalHello OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The interval between EGP Hello command
                retransmissions (in hundredths of a second).  This
                represents the t1 timer as defined in RFC 904."
        ::= { egpNeighEntry 12 }

egpNeighIntervalPoll OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The interval between EGP poll command
                retransmissions (in hundredths of a second).  This
                represents the t3 timer as defined in RFC 904."
        ::= { egpNeighEntry 13 }

egpNeighMode OBJECT-TYPE
        SYNTAX  INTEGER { active(1), passive(2) }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The polling mode of this EGP entity, either
                passive or active."
        ::= { egpNeighEntry 14 }
```

```
egpNeighEventTrigger OBJECT-TYPE
        SYNTAX  INTEGER { start(1), stop(2) }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "A control variable used to trigger operator-
                initiated Start and Stop events.  When read, this
                variable always returns the most recent value that
                egpNeighEventTrigger was set to.  If it has not
                been set since the last initialization of the
                network management subsystem on the node, it
                returns a value of 'stop'.

                When set, this variable causes a Start or Stop
                event on the specified neighbor, as specified on
                pages 8-10 of RFC 904.  Briefly, a Start event
                causes an Idle peer to begin neighbor acquisition
                and a non-Idle peer to reinitiate neighbor
                acquisition.  A stop event causes a non-Idle peer
                to return to the Idle state until a Start event
                occurs, either via egpNeighEventTrigger or
                otherwise."
        ::= { egpNeighEntry 15 }
```

Additional EGP Objects

```
egpAs OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The autonomous system number of this EGP entity."
        ::= { egp 6 }
```

The Transmission Group

```
-- Based on the transmission media underlying each interface
-- on a system, the corresponding portion of the Transmission
-- group is mandatory for that system.

-- When Internet-standard definitions for managing
-- transmission media are defined, the transmission group is
-- used to provide a prefix for the names of those objects.

-- Typically, such definitions reside in the experimental
-- portion of the MIB until they are "proven", then as a
-- part of the Internet standardization process, the
-- definitions are accordingly elevated and a new object
-- identifier, under the transmission group is defined. By
-- convention, the name assigned is:
--
--     type OBJECT IDENTIFIER    ::= { transmission number }
--
-- where "type" is the symbolic value used for the media in
-- the ifType column of the ifTable object, and "number" is
-- the actual integer value corresponding to the symbol.
```

The SNMP Group

```
-- Implementation of the SNMP group is mandatory for all
-- systems which support an SNMP protocol entity.  Some of
-- the objects defined below will be zero-valued in those
-- SNMP implementations that are optimized to support only
-- those functions specific to either a management agent or
-- a management station.  In particular, it should be
-- observed that the objects below refer to an SNMP entity,
-- and there may be several SNMP entities residing on a
-- managed node (e.g., if the node is hosting acting as
-- a management station).

snmpInPkts OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of Messages delivered to the
                SNMP entity from the transport service."
        ::= { snmp 1 }
snmpOutPkts OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of SNMP Messages which were
                passed from the SNMP protocol entity to the
                transport service."
        ::= { snmp 2 }

snmpInBadVersions OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of SNMP Messages which were
                delivered to the SNMP protocol entity and were for
                an unsupported SNMP version."
        ::= { snmp 3 }

snmpInBadCommunityNames OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
        "The total number of SNMP Messages delivered to
        the SNMP protocol entity which used a SNMP
        community name not known to said entity."
        ::= { snmp 4 }
```

```
snmpInBadCommunityUses OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of SNMP Messages delivered to
                the SNMP protocol entity which represented an SNMP
                operation which was not allowed by the SNMP
                community named in the Message."
        ::= { snmp 5 }

snmpInASNParseErrs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of ASN.1 or BER errors
                encountered by the SNMP protocol entity when
                decoding received SNMP Messages."
        ::= { snmp 6 }

-- { snmp 7 } is not used

snmpInTooBigs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of SNMP PDUs which were
                delivered to the SNMP protocol entity and for
                which the value of the error-status field is
                'tooBig'."
        ::= { snmp 8 }

snmpInNoSuchNames OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of SNMP PDUs which were
                delivered to the SNMP protocol entity and for
                which the value of the error-status field is
                'noSuchName'."
        ::= { snmp 9 }
snmpInBadValues OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of SNMP PDUs which were
                delivered to the SNMP protocol entity and for
                which the value of the error-status field is
                'badValue'."
        ::= { snmp 10 }

snmpInReadOnlys OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number valid SNMP PDUs which were
                delivered to the SNMP protocol entity and for
                which the value of the error-status field is
                'readOnly'.  It should be noted that it is a
                protocol error to generate an SNMP PDU which
                contains the value 'readOnly' in the error-status
                field, as such this object is provided as a means
                of detecting incorrect implementations of the
                SNMP."
        ::= { snmp 11 }

snmpInGenErrs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of SNMP PDUs which were
                delivered to the SNMP protocol entity and for
                which the value of the error-status field is
                'genErr'."
                ::= { snmp 12 }

snmpInTotalReqVars OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                The total number of MIB objects which have been
                retrieved successfully by the SNMP protocol entity
                as the result of receiving valid SNMP Get-Request
                and Get-Next PDUs."
        ::= { snmp 13 }

snmpInTotalSetVars OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of MIB objects which have been
                altered successfully by the SNMP protocol entity
                as the result of receiving valid SNMP Set-Request
                PDUs."
        ::= { snmp 14 }
```

```
snmpInGetRequests OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of SNMP Get-Request PDUs which
                have been accepted and processed by the SNMP
                protocol entity."
        ::= { snmp 15 }

snmpInGetNexts OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of SNMP Get-Next PDUs which have
                been accepted and processed by the SNMP protocol
                entity."
        ::= { snmp 16 }

snmpInSetRequests OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of SNMP Set-Request PDUs which
                have been accepted and processed by the SNMP
                protocol entity."
        ::= { snmp 17 }

snmpInGetResponses OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of SNMP Get-Response PDUs which
                have been accepted and processed by the SNMP
                protocol entity."
        ::= { snmp 18 }

snmpInTraps OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of SNMP Trap PDUs which have
                been accepted and processed by the SNMP protocol
                entity."
        ::= { snmp 19 }

snmpOutTooBigs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of SNMP PDUs which were
                generated by the SNMP protocol entity and for
                which the value of the error-status field is
                'tooBig.'"
        ::= { snmp 20 }

snmpOutNoSuchNames OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of SNMP PDUs which were
                generated by the SNMP protocol entity and for
                which the value of the error-status is
                'noSuchName'."
        ::= { snmp 21 }

snmpOutBadValues OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of SNMP PDUs which were
                generated by the SNMP protocol entity and for
                which the value of the error-status field is
                'badValue'."
        ::= { snmp 22 }

-- { snmp 23 } is not used

snmpOutGenErrs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of SNMP PDUs which were
                generated by the SNMP protocol entity and for
                which the value of the error-status field is
                'genErr'."
        ::= { snmp 24 }

snmpOutGetRequests OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of SNMP Get-Request PDUs which
                have been generated by the SNMP protocol entity."
        ::= { snmp 25 }
```

```
snmpOutGetNexts OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of SNMP Get-Next PDUs which have
                been generated by the SNMP protocol entity."
        ::= { snmp 26 }

snmpOutSetRequests OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of SNMP Set-Request PDUs which
                have been generated by the SNMP protocol entity."
        ::= { snmp 27 }

snmpOutGetResponses OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of SNMP Get-Response PDUs which
                have been generated by the SNMP protocol entity."
        ::= { snmp 28 }

snmpOutTraps OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The total number of SNMP Trap PDUs which have
                been generated by the SNMP protocol entity."
        ::= { snmp 29 }

snmpEnableAuthenTraps OBJECT-TYPE
        SYNTAX  INTEGER { enabled(1), disabled(2) }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "Indicates whether the SNMP agent process is
                permitted to generate authentication-failure
                traps.  The value of this object overrides any
                configuration information; as such, it provides a
                means whereby all authentication-failure traps may
                be disabled.

                Note that it is strongly recommended that this
                object be stored in non-volatile memory so that it
                remains constant between re-initializations of the
                network management system."

        ::= { snmp 30 }

END
```