Network Working Group
Request for Comments: 3606
Category: Standards Track

F. Ly
Pedestal Networks
M. Noto
Cisco Systems
A. Smith
Consultant
E. Spiegel
Cisco Systems
K. Tesink
Telcordia Technologies

November 2003

Definitions of Supplemental Managed Objects for ATM Interface

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Abstract

This memo defines objects used for managing ATM-based interfaces, devices, and services, in addition to those defined in RFC 2515, the ATM-MIB, to provide additional support for the management of ATM Switched Virtual Connections (SVCs) and ATM Permanent Virtual Connections (PVCs).

Table of Contents

1. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIv2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

2. Overview

The purpose of this memo is to provide additional capabilities, not found in the ATM-MIB [RFC2515], which are needed to manage ATM interfaces. This memo addresses the following areas:

- ATM Switch Support
- ATM Service Support
- ATM Host Support

In addition, this memo also provides ATM trap support.

2.1. Background

In addition to the MIB module defined in this memo, other MIB modules are necessary to manage ATM interfaces, links and cross-connects. Examples include MIB II for general system and interface management ([RFC2863]), the DS3 ([RFC2496]) or SONET MIBs ([RFC3592]) for management of SONET and DS3 physical interfaces, and, as appropriate, MIB modules for applications that make use of ATM, such as SMDS [RFC1694] and LAN Emulation [ATM Forum LANE]. These MIB modules are outside the scope of this specification.

This MIB module also requires the use of the ATM-MIB module defined in [RFC2515] and ATM-specific textual conventions defined in [RFC2514].

ATM Endpoint applications such as ATM LAN Emulation or Classical IP-over-ATM Clients and Servers use ATM to establish SVC/PVC connections for exchanging control and data information. The agents of these ATM applications must provide the network manager with information on the SVC/PVCs in use and which applications are using them. The information can be made generic so as to apply to all ATM applications. This memo defines extensions to the ATM-MIB [RFC2515] in order to support this.

The current specification of this supplemental ATM2-MIB is based on SNMPv2 SMI.

2.2. Important Definitions

The following terms are defined here and used throughout this MIB:

- Virtual Path Link (VPL)
- Virtual Path Connection (VPC)
- Virtual Path Segment (VP Segment)
- Virtual Channel Link (VCL)
- Virtual Channel Connection (VCC)
- Virtual Channel Segment (VC Segment).

The figures on the next page show how these terms apply in typical ATM network topologies. Additional terms relevant to this MIB are defined and illustrated in the ATM Terminology section 3 of [RFC2515].

```
    _____      _______      _______      _______      _____
   |     |____|       |____|       |____|       |____|     |
   |Host1|    |SwitchA|    |SwitchB|    |SwitchC|    |Host2|
   |     |____|       |____|       |____|       |____|     |
   |_____|    |_______|    |_______|    |_______|    |_____|

         |<-->|                    |<-->|
    Virtual Path Link         Virtual Path Link

         |<----------------------------------------->|
                     Virtual Path Connection
                    (between Host1 and Host2)

                      |<--------------->|
                      Virtual Path Segment
                 (between SwitchA and SwitchC)
```

Figure 1: Examples of Virtual Path Links, Virtual Path
Connection, and Virtual Path Segment

```
    _____       _______      _______      _______      _____
   |     |____|       |____|       |____|       |____|     |
   |Host1|----|SwitchA|----|SwitchB|----|SwitchC|----|Host2|
   |     |____|       |____|       |____|       |____|     |
   |_____|    |_______|    |_______|    |_______|    |_____|

         |<-->|                    |<-->|
   Virtual Channel Link      Virtual Channel Link

         |<----------------------------------------->|
                   Virtual Channel Connection
                   (between Host1 and Host2)

                      |<--------------->|
                    Virtual Channel Segment
                 (between SwitchA and SwitchC)
```

Figure 2: Examples of Virtual Channel Links, Virtual Channel Connection, and Virtual Channel Segment

3. Conventions used in the MIB

3.1. Structure

The managed ATM objects are arranged as follows:

Table	Host	Switch	Service
atmSvcVcCrossConnectTable		Y	Y
atmSvcVpCrossConnectTable		Y	Y
atmSigStatTable	Y	Y	Y
atmSigSupportTable		Y	Y
atmSigDescrParamTable	Y		
atmIfRegisteredAddrTable		Y	Y
atmVclAddrTable	Y		
atmAddrVclTable	Y		
atmVplStatTable	Y	Y	Y
atmVplLogicalPortTable	Y	Y	Y
atmVclStatTable	Y	Y	Y
atmAal5VclStatTable	Y		
atmVclGenTable	Y		
atmInterfaceExtTable	Y	Y	Y
atmIlmiSrvcRegTable		Y	Y
atmIlmiNetworkPrefixTable		Y	Y
atmSwitchAddressTable		Y	
atmVpCrossConnectXTable			Y
atmVcCrossConnectXTable			Y
atmCurrentlyFailingPVplTable	Y	Y	Y
atmCurrentlyFailingPVclTable	Y	Y	Y

Table 1: MIB structure

3.1.1. ATM SVC VP Cross-Connect Table

This table provides the SVC VP Cross-Connect (SVPC) information. The equivalent PVC VP Cross-Connect table is defined in [RFC2515]. This table also includes cross-connect information for Soft PVPs.

This table contains configuration and state information of all SVC VP point-to-point, point-to-multipoint, or multipoint-to-multipoint VP cross-connects.

This table has read-only access and can be used to monitor the cross-connects which connect the VPLs together in an ATM switch or network. The atmSvcVpCrossConnectIndex is used to associate the related SVC VPLs that are cross-connected together. The atmSvcVpCrossConnectRowStatus object has read-write access to allow for tear-down.

The ATM SVC VP Cross-Connect Table models each bi-directional Switched Virtual Circuit (SVC) VP cross-connect as a set of entries in the atmSvcVpCrossConnectTable. A point-to-point VPC cross-connect is modeled as one entry; a point-to-multipoint (N leafs) VPC cross-connect as N entries in this table; and a multipoint-to-multipoint (N parties) VPC cross-connect as N(N-1)/2 entries in this table. In the latter cases, all the N (or N(N-1)/2) entries are associated with a single VPC cross-connect by having the same value of atmSvcVpCrossConnectIndex.

```
     _________________________________________
     |                                       |
 Low |         ATM Switch or Network         | High
 port|                                       | port
_____|>> from low to high VPC traffic flow >>|______
     |<< from high to low VPC traffic flow <<|
     |_______________________________________|
```

Figure 3: VPC Cross-Connect Model

The terms low and high are chosen to represent numerical ordering of the two interfaces associated with a VPC cross-connect. That is, the ATM interface with the lower value of ifIndex is termed ’low’, while the other ATM interface associated with the VPC cross-connect is termed ’high’.

3.1.2. ATM SVC VC Cross-Connect Table

This table provides the SVC Cross-Connect (SVCC) information. The equivalent PVC VC Cross-Connect table is defined in [RFC2515]. This table also includes cross-connect information for Soft PVCs.

This table is used to model a bi-directional point-to-point, point-to-multipoint or multipoint-to-multipoint SVC VC cross-connect.

This table has read-only access and is used to monitor the cross-connects which connect the VCLs together in an ATM switch or network that belong to a VC connection. The atmSvcVcCrossConnectIndex is used to associate the related SVC VCLs that are cross-connected together. The atmSvcVcCrossConnectRowStatus object has read-write access to allow for tear-down.

The ATM SVC VC Cross-Connect Table models each bi-directional Switched Virtual Circuit (SVC) VC cross-connect as a set of entries in the atmSvcVcCrossConnectTable. A point-to-point VC cross-connect is modeled as one entry; a point-to-multipoint (N leafs) VC cross-connect as N entries in this table; and a multipoint-to-multipoint (N parties) VPC cross-connect as N(N-1)/2 entries in this table. In the latter cases, all the N (or N(N-1)/2) entries are associated with a single VPC cross-connect by having the same value of atmSvcVcCrossConnectIndex.

```
              ______________________________________
             |                                      |
     Low     |         ATM Switch or Network        |  High
     port    |                                      |  port
    _________|>> from low to high VC traffic flow >>|______
             |<< from high to low VC traffic flow <<|
             |______________________________________|
```

Figure 4: VC Cross-Connect Model

The terms low and high are chosen to represent numerical ordering of the two interfaces associated with a VPC cross-connect. That is, the ATM interface with the lower value of ifIndex is termed 'low', while the other ATM interface associated with the VPC cross-connect is termed 'high'.

3.1.3. ATM Interface Signalling Statistics Table

This table provides statistical information of the signalling entity. A signalling entity can be deployed over an ATM interface as defined in the atmInterfaceConfTable [RFC2515], a logical ATM interface defined in section 5.1.10.1 in this document, or a proprietary virtual interface as described in the atmInterfaceExtTable. To monitor the signalling entity, a few counters are provided. They are defined as:

atmSigSSCOPConEvents
atmSigSSCOPErrdPdus
atmSigDetectSetupAttempts
atmSigEmitSetupAttempts
atmSigDetectUnavailRoutes

atmSigEmitUnavailRoutes
atmSigDetectUnavailResrcs
atmSigEmitUnavailResrcs
atmSigDetectCldPtyEvents
atmSigEmitCldPtyEvents
atmSigDetectMsgErrors
atmSigEmitMsgErrors
atmSigDetectClgPtyEvents
atmSigEmitClgPtyEvents
atmSigDetectTimerExpireds
atmSigEmitTimerExpireds
atmSigDetectRestarts
atmSigEmitRestarts
atmSigInEstabls
atmSigOutEstabls

3.1.4. ATM Signalling Capability Support

A number of Information Elements may or may not be supported by ATM switches or ATM Services. Hence, for trouble isolation it is important to keep track which particular Information Elements are supported. The corresponding group of objects must be supported by switches or services supporting SVCs, and indicate whether the following Information Elements are enabled/disabled:

1) Calling party number

2) Calling party subaddress

3) Called party subaddress

4) Broadband high layer information

5) Broadband low layer information

6) Broadband Repeat Indicator

7) AAL parameters

The last parameter, Preferred Carrier Pre-Subscription, identifies the carrier to which intercarrier calls originated from this interface are routed when transit network selection information is not provided by the calling party.

3.1.5. Signalling Descriptor Parameter Table

This table extends the ATM VCL table of the ATM-MIB [RFC2515] to include all other necessary signalling information as specified in the ATM Forum UNI Specifications [ATM Forum 3.0] and [ATM Forum UNI 3.1]. A user can create an entry with all signalling parameters and later use that entry to specify the signalling characteristics of SVCs.

Some of the signalling parameters, such as the AAL5 parameters information element, are reflected in the VCL and VPL tables, and this table only contains the remaining AAL5 parameters.

Signalling attributes can be grouped into following categories:

1) ATM Adaptation Layer Parameters

   Information in this group is captured in the ATM Signalling Descriptor Parameter Table defined in this memo. Please refer to section 5.4.5.5 of [ATM Forum 3.0] and [ATM Forum UNI 3.1].

2) Broadband High Layer Information

   Information in this group is captured by the ATM Signalling Descriptor Parameter Table defined in this memo. Please refer to section 5.4.5.8 of [ATM Forum 3.0] and [ATM Forum UNI 3.1].

3) Broadband Low Layer Information

   Information in this group is captured by the ATM Signalling Descriptor Parameter Table defined in this memo. Please refer to section 5.4.5.9 of [ATM Forum 3.0] and [ATM Forum UNI 3.1].

3.1.6. ATM Interface Registered Address Table

This table contains a list of ATM addresses that can be used for calls to and from a given interface by a switch or service. The ATM addresses are either registered by the endsystem via ILMI or statically configured. This table does not expose PNNI reachability information. This table only applies to switches and network services. See also Section 5.2.

3.1.7. ATM VPI/VCI to Address Mapping Table

In the atmVclAddrTable, the object atmVclAddrAddr can either be an ATM Local Address or an ATM Remote Address which represent the two endpoint addresses of a VCL. ATM Local Address identifies the local endpoint of the VCL represented by this agent. The ATM Remote address represents the address of the ATM application at the other end of the VCL.

3.1.8. ATM Address to VPI/VCI Mapping Table

This table provides an alternative way to retrieve the atmVclTable. This table can be used to retrieve the indexing to the atmVclTable by an ATM address.

3.1.9. ATM VPL Statistics Table

The atmVplStatTable includes per-VPL cell counters. The VPL cell counters count the valid ATM cells. The valid ATM cells include the user and OAM cells but exclude the physical layer (e.g., idle cells) and unassigned cells. Cells coming into an ATM managed system are counted differently with the high Cell Loss Priority (CLP=0) or low Cell Loss Priority (CLP=1). The cells are tagged, passed or discarded depending on the incoming CLP value and the policed cell rate by the "traffic policing" entity in the ATM managed system. Refer to [ATM Forum 3.0] and [ATM Forum UNI 3.1] for a description of the traffic policing.

In the switch where the traffic policing is not supported, cells are passed or discarded depending on the bandwidth and buffering capacity of the switching fabric. The Output Tagged Cells counter, in this case, is always zero.

```
                 _______________
                 | ATM Managed |
      Input      | System      | Output
      CLP=0 cells|             | CLP=0 cells
      ---------->|             |----------->
      CLP=1 cells| (traffic    | CLP=1 cells
      ---------->| policing    |----------->
                 | entity)     | Tagged cells (CLP=1)
                 |_____________|----------->
                  |Discard  | Discard
                  |CLP=0    | CLP=1
                  |cells    | cells
                  |         |
                  V         V
```

Figure 5: ATM Cell Counters per VPL

In this table, cells coming into and out of the managed ATM system are counted as the total number of cells and the cells with the CLP=0. The CLP=1 counter is derived by subtracting CLP=0 cells from the total cells. In addition, cells that are tagged on the output are also counted. The output CLP=1 cells equals the total cells out count minus both the CLP=0 cells and the tagged cells.

3.1.10. ATM VPL Logical Port Table

The ATM VPL Logical Port Table includes all ATM logical port interface configuration information.

3.1.10.1. ATM Logical Port Interface

The interface type "ATM Logical Port" (ifType=80) is defined to allow the representation of a VP Tunnel, which is a VPL used as a trunk connection (most likely between devices that are not physically adjacent), providing for multiplexing and demultiplexing of VCs on the VP. Figure 6 illustrates such a VP Tunnel.

Note: the "ATM Logical Port" interface is more of a logical port, compared with an interface of type "ATM" which is more of a physical port that provides for the transport of many VP and VC connections between adjacent devices.

```
                       <------VP Tunnel------>
                 ATM Switch A             ATM Switch B
                ------------             -----------
                |ATM       |_____________|ATM       |
                |X-Connect |      .      |X-Connect |
         VCL1   |Point     | VPL1 . VPL2 |Point     |  VCL4
      O---------|----X-----|----- . -----|----X-----|-----O
                |    X-----|----- . -----|----X     |
                |    |     |_____________|    |     |
                ------------             ------------
                     | VCL2                    | VCL3
                     O                         O
```

Figure 6: Virtual Path Tunnel

In Figure 6, a VP tunnel (denoted as VPL1 by Switch A, and as VPL2 by Switch B) is used to connect VCL1 with VCL4 and VCL2 with VCL3. Figure 6 shows only one VP tunnel, but there can be multiple VP tunnels over the same physical interface.

A particularly useful VP tunnel scenario is tunneling across a public network that does not support signalling. In Figure 6 above, assume Switches A and B are private switches that signal over the VP to set up connections transparently through the public network. The public network would just transport a PVC VP between the two switches.

Because the VP Tunnel constitutes an interface between two ATM devices that are not necessarily physically adjacent, most of the management information pertaining to the interface may differ for the tunnel, including:

- active VPI/VCI fields (the tunnel may be a subset of the parent interface).
- maximum number of VCCs
- configured VCCs
- ILMI VPI/VCI values
- ATM address type
- ATM administrative address
- received/transmitted cells.

3.1.10.2. How to create an ATM Logical Port interface

On ATM devices supporting VP tunnels, the ATM Logical Port Interface Table can be used to create VP tunnels. To create an ATM Logical Port interface via SNMP:

- Create a VPL (e.g., VPI=a on an existing ATM interface which has ifIndex=x) in the atmVplTable.

- Set the object atmVplLogicalPortDef to isLogicalIf. A new row in the ifTable is then created by the agent, with ifIndex=y, to represent the ATM Logical Port interface. The object atmVplLogicalPortIndex is also set to y by the agent to represent the ifIndex value of the ATM Logical Port interface.

- The ifEntry values are set for the ATM Logical Port interface (ifIndex=y) as discussed in RFC 2515, with the following exceptions:
  * ifType - a new enumerated value of atmLogical(80) was added to IANAifType, specifying an "ATM Logical Port" interface.
  * ifSpeed - The total bandwidth in bits per second for use by the ATM layer. Computed from the traffic descriptor for the VPL.

```
                * ifOperStatus - determined hierarchically,
                  depending on the state of the physical
                  atm-cell layer interface beneath it,
                  and the ILMI on the VP.
             * ifInOctets, ifOutOctets - support of
               these objects is not mandatory for ATM
               Logical Port interfaces.
             * ifInErrors - always zero, HEC errors are
               specified for the atm cell-layer interface
               beneath it.
             * ifInUnknownProtos - always zero, errors
               are specified for the atm cell-layer
               interface beneath it.

       - The atmInterfaceConfEntry values are set and reported
         as discussed in [RFC2515], with the following exceptions:
             * atmInterfaceMaxVpcs - 0.
             * atmInterfaceConfVpcs - 0.
             * atmInterfaceIlmiVpi - VPI of the VP tunnel.

       - The atmInterfaceExtEntry values are set and reported
         as follows:
             * atmInterfaceConfMaxSvpcVpi - VPI of the VP tunnel,
               although VPCs cannot be setup on a VP tunnel.
             * atmInterfaceCurrentMaxSvpcVpi - VPI of VP tunnel,
               although VPCs cannot be setup on a VP tunnel.
             * atmInterfaceConfMaxSvccVpi - VPI of the VP tunnel.
             * atmInterfaceCurrentMaxSvccVpi - VPI of VP tunnel.
             * atmIntfPvcFailures - Includes failures of PVCLs
               within the VP tunnel, but not of the PVPL itself,
               since those are reported on the atm(37) interface.
             * atmIntfCurrentlyFailingPVpls - 0.
             * atmIntfPvcFailuresTrapEnable - Enables traps for
               PVCL failures within the VP tunnel, but not for
               the PVPL itself, since the latter are generated on
               behalf of the atm(37) interface.

       - An entry is created in the ifStackTable, with
         values: ifStackHigherLayer=y, ifStackLowerLayer=x.

       - VCLs defined on the VP tunnel are indexed by
         ifIndex=y, VPI=a, VCI.
```

3.1.11. ATM VCL Statistics Table

The atmVclStatTable includes per-VCL cell counters. The VCL cell counters count the valid ATM cells. The valid ATM cells include the user and OAM cells but exclude the physical layer (e.g., idle cells) and unassigned cells. Cells coming into an ATM managed system are counted differently with the high Cell Loss Priority (CLP=0) or low Cell Loss Priority (CLP=1). The cells are tagged, passed or discarded depending on the incoming CLP value and the policed cell rate by the "traffic policing" entity in the ATM managed system. Refer to [ATM Forum 3.0] and [ATM Forum UNI 3.1] for the description of the traffic policing.

In a switch where the traffic policing is not supported, cells are passed or discarded depending on the bandwidth and buffering capacity of the switching fabric. The Output Tagged Cells counter, in this case, is always zero.

```
                 _______________
                 | ATM Managed |
      Input      | System      | Output
      CLP=0 cells|             | CLP=0 cells
      ---------->|             |----------->
      CLP=1 cells| (traffic    | CLP=1 cells
      ---------->| policing    |----------->
                 | entity)     | Tagged cells (CLP=1)
                 |_____________|----------->
                  |Discard  | Discard
                  |CLP=0    | CLP=1
                  |cells    | cells
                  |         |
                  V         V
```

Figure 7: ATM Cell Counters per VCL

In this table, cells coming into and out of the managed ATM system are counted as the total number of cells and the cells with the CLP=0. The CLP=1 counter is derived by subtracting CLP=0 cells from the total cells. In addition, cells that are tagged on the output are also counted. The output CLP=1 cells equals the total cells out count minus both the CLP=0 cells and the tagged cells.

3.1.12. ATM VC General Information Table

This table contains the general information for each VC. It provides an index to the atmSigDescrParamTable defined in this MIB. This table is an extension to the atmVclTable defined in the ATM-MIB [RFC2515].

3.1.13. ATM Interface Configuration Extension Table

The ATM Interface Configuration Extension Table contains ATM interface information that supplements the atmInterfaceConfTable defined in [RFC2515]. It includes the configuration information of the interface type (i.e., connection setup procedures) and ILMI.

A network manager can configure the interface to run a specific type of connection setup procedures (i.e., protocol and version) such as ITU-T DSS2, ATM Forum UNI 3.1, PNNI 1.0 or BICI 2.0. It can also dictate the role of the managed entity as one side of the interface. For example, if an interface is configured to run ATM Forum UNI 3.1, the managed entity has to be told to run as either the network side or the user side of the UNI.

The objects atmIntfConfigType and atmIntfConfigSide are used for configuration and the objects atmIntfActualType and atmIntfActualSide are used for reading back the actual interface protocol and version.

The following table describes all the valid combinations of configuration of the interface type and side. Note that the value N/A meaning not applicable, should be set to the value other(1) when used.

atmIntfConfigType	atmIntfConfigSide
autoConfig	N/A
ituDss2	user/network
atmfUni3Dot0	user/network
atmfUni3Dot1	user/network
atmfUni4Dot0	user/network
atmfIispUni3Dot0	user/network
atmfIispUni3Dot1	user/network
atmfIispUni4Dot0	user/network
atmfPnni1Dot0	N/A
atmfBici2Dot0	N/A
atmfUniPvcOnly	user/network
atmfNniPvcOnly	N/A

When the value of the object atmIntfConfigType is configured to autoConfig(2), the interface type is determined via the ATM Forum ILMI auto-configuration procedures [ATM Forum ILMI]. There is no need to set the interface side since it should be a derived value. The PNNI and BICI interfaces are always symmetric so setting the interface side is also not necessary.

This table also includes the configured and negotiated maximum VPI value per ATM interface, and the configured and negotiated minimum VCI value per ATM interface. Refer to [ATM Forum ILMI] Sections 8.2.3.8 through 8.2.3.10 for a detailed description.

The following figure provides an example how the current minimum VCI values are derived from the configured minimum VCI values and the neighboring minimum VCI values:

```
+--------+              +--------+              +--------+
|  ATM   | ifA      ifB |  ATM   | ifC      ifD |  ATM   |
| Device |--------------| Device |--------------| Device |
+--------+              +--------+              +--------+


     ifA:  Configured Min SVCC VCI = 32  (configured)
           Current Min SVCC VCI    = 40  (negotiated)

     ifB:  Configured Min SVCC VCI = 40  (configured)
           Current Min SVCC VCI    = 40  (negotiated)

     ifC:  Configured Min SVCC VCI = 32  (configured)
           Current Min SVCC VCI    = 32  (negotiated)

     ifD:  Configured Min SVCC VCI = 32  (configured)
           Current Min SVCC VCI    = 32  (negotiated)
```

Between ifA and ifB, the maximum of the two vales for atmInterfaceConfMinSvccVci is 40, so both interfaces set their atmInterfaceCurrentMinSvccVci values to 40. On the other hand, since ifC and ifD both are configured with atmInterfaceConfMinSvccVci values of 32, they set their atmInterfaceCurrentMinSvccVci values to 32.

Figure 8: Examples of configured vs. negotiated ILMI values

3.1.14. ATM ILMI Service Registry Table

This table contains information used by the switch/service to inform ATM hosts of the location of ATM network services such as the LAN Emulation Configuration Server (LECS), the ATM Name Server (ANS), the ATMARP Server, the Multicast Address Resolution Server (MARS), and the NHRP Server (NHS). Entries in this table are exported to adjacent devices via ILMI over either all or a few user selected ATM interfaces.

As an example, let’s assume that:

- An ATM switch X has three interfaces if1, if2 and if3.
- There are two ATM network services offered, a1.a2...aN and b1.b2...bN, where a1.a2...aN is an object identifier used to identify the first service, and b1.b2...bN is the object identifier for the other service.
- The first service is available at the ATM address ’a’.
- The second service is available at the ATM address ’b’.
- The first service can be used by any device connecting to the switch X.
- The second service can be used only by devices that connect to interfaces if1 and if3 on switch X.

```
     +------------------+    +------------------+
     |service a1.a2...aN|    |service b1.b2...bN|
     |                  |    |                  |
     | ATM address = a  |    | ATM address = b  |
     +--------+---------+    +--------+---------+
              |                       |
              |                       |
     +--------+-----------------------+---------+
     |             ATM NETWORK                  |
     +-----------------+------------------------+
                       |
                       |
                +-------------+
                |  switch X   |
                +-+----+----+-+
                  |    |    |
                  |    |    |
                 if1  if2  if3   (interfaces)
```

Figure 9: ATM topology with registered services

The table for switch X will contain three entries:

- one entry for the "a1.a2...aN", implicitly available to any devices on switch X.
- two entries for the "b1.b2...bN" (one for each interface where this service can be explicitly used).

The content of the table is:

- Service Identifier:	a1.a2...aN	b1.b2...bN	b1.b2...bN
- ATM address:	a	b	b
- Arbitrary index:	m	n	p
- Available interface:	0	1	3

where the Service Identifier values a1.a2...aN and b1.b2...bN are represented by atmIlmiSrvcRegServiceID, the ATM addresses a and b are represented by atmIlmiSrvcRegATMAddress, the values m, n, and p are arbitrary non-zero integer parameters (necessary in this example to differentiate the two entries for b1.b2...bN that are both available at the ATM address ’b’) represented by atmIlmiSrvcRegAddressIndex, and the available interfaces are represented by atmIlmiSrvcRegIndex, where the special value 0 indicates any ATM interface.

When querying the ILMI service registry table, through the ILMI protocol:

- the device attached to interface if1 will obtain the address a and b.

- the device attached to interface if2 will obtain the address a only.

- the device attached to interface if3 will obtain the address a and b.

3.1.15. ILMI Network Prefix Table

A table specifying per-interface network prefix(es) supplied by the network side of the UNI during ILMI address registration. When no network prefixes are specified for a particular interface, one or more network prefixes based on the switch address(es) may be used for ILMI address registration.

3.1.16. ATM Switch Address Table

This table contains one or more ATM endsystem addresses on a per-switch basis. These addresses are used to identify the switch. When no ILMI network prefixes are configured for certain interfaces, network prefixes based on the switch address(es) may be used for ILMI address registration.

3.1.17. AAL5 per-VCC Statistics Table

This table contains the AAL5 statistics for the VCCs.

3.1.18. ATM VP Cross-Connect Extension Table

This table extends the atmVpCrossConnectTable defined in ATM-MIB [RFC2515].

3.1.19. ATM VC Cross-Connect Extension Table

This table extends the atmVcCrossConnectTable defined in ATM-MIB [RFC2515].

3.1.20. Currently Failing PVPL Table

This table contains all the PVPLs that are in trouble.

3.1.21. Currently Failing PVCL Table

This table contains all the PVCLs that are in trouble.

3.1.22. Leaf Initiated Join Counter support

Two counter objects are added to count the number of leaf intiated setup requests and setup failures.

3.2. Network and User Addresses

At the user side of a given ATM UNI interface there may be an address, "ifPhysAddress", to identify the interface. In addition, there may be several other addresses which can be used to originate and receive calls. These other addresses that are used to receive calls are listed in the "ifRcvAddrTable" defined in RFC 2863. The registered addresses on the network side are listed in the ATM Registered Address Table. The ATM Registered Address Table is supported by switches and network services. It is not supported by hosts.

3.3. Configuration of VPLs, VCLs, and Cross-Connects

The ATM Managed Objects needed to support the configuration of VPLs, VCLs, and Cross-Connects of the Permanent VPLs and VCLs are defined in the ATM-MIB [RFC2515]. Cross-Connects of the Switched VPLs and VCLs are defined in this memo.

3.4. ATM-related Trap Support

Traps are defined to detect changes in the status of permanent VPLs and VCLs. The current up/down status of each permanent VPL or VCL is indicated by the atmVplOperStatus or atmVclOperStatus object, respectively. Several tables and objects and one trap are defined in order to help network managers quickly and efficiently detect changes in the status of permanent virtual links. Through use of these tables, objects, and traps, the time consuming and resource intensive task of continuously polling each row in the entire atmVplTable and atmVclTable can be avoided.

The atmIntfPvcFailures counter and the atmIntfCurrentlyFailingPVpls and atmIntfCurrentlyFailingPVcls gauges provide a quick means of determining the status of all PVPLs and PVCLs on an interface. The atmCurrentlyFailingPVplTable and the atmCurrentlyFailingPVclTable list all of the problematic PVPLs and PVCLs, respectively, allowing them to be quickly identified.

The atmIntfPvcFailuresTrap is generated just after a PVPL or PVCL on a particular interface leaves the 'up' operational state. Managers can then determine which PVPLs and/or PVCLs are failing by reading the atmCurrentlyFailingPVplTable and the atmCurrentlyFailingPVclTable. Generation of the atmIntfPvcFailuresTrap is rate limited by suppressing all traps that would occur within atmIntfPvcNotificationInterval of a previous trap for the same interface. Managers should continuously poll the tables and objects mentioned above for at least this amount of time in order to keep up with the state of the network.

4. Conformance and Compliance

See the conformance and compliance statements within the information module.

5. Definitions

```
ATM2-MIB DEFINITIONS ::= BEGIN

   IMPORTS
       MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
       Gauge32, Counter32, Integer32
           FROM SNMPv2-SMI
       TruthValue, RowStatus, TimeStamp
           FROM SNMPv2-TC
       MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
           FROM SNMPv2-CONF
       SnmpAdminString
           FROM SNMP-FRAMEWORK-MIB
       InterfaceIndex, InterfaceIndexOrZero, ifIndex
           FROM IF-MIB
       atmMIBObjects, atmInterfaceConfEntry,
       atmVplEntry, atmVplVpi,
       atmVclEntry, atmVclVpi, atmVclVci,
       atmVpCrossConnectEntry, atmVcCrossConnectEntry
           FROM ATM-MIB
       AtmAddr, AtmSigDescrParamIndex,
       AtmInterfaceType, AtmIlmiNetworkPrefix,
       AtmVcIdentifier, AtmVpIdentifier,
       AtmTrafficDescrParamIndex
           FROM ATM-TC-MIB;

   atm2MIB MODULE-IDENTITY
       LAST-UPDATED "200309230000Z"
       ORGANIZATION "IETF AToMMIB Working Group"
       CONTACT-INFO
         "AToMMIB WG
             http://www.ietf.org/html.charters/atommib-charter.html
           Editors:
                    Faye Ly
           Postal:  Pedestal Networks
                    6503 Dumbarton Circle
                    Fremont, CA 94555
                    USA
           Tel:     +1 510 896 2908
           E-Mail:  faye@pedestalnetworks.com

                    Michael Noto
           Postal:  Cisco Systems
                    170 W. Tasman Drive
                    San Jose, CA 95134-1706
                    USA

           E-mail:  mnoto@cisco.com

                    Andrew Smith
           Postal:  Consultant

           E-Mail:  ah_smith@acm.org

                    Ethan Mickey Spiegel
           Postal:  Cisco Systems
                    170 W. Tasman Drive
                    San Jose, CA 95134-1706
                    USA
           Tel:     +1 408 526 6408
           Fax:     +1 408 526 6488
           E-Mail:  mspiegel@cisco.com

                    Kaj Tesink
           Postal:  Telcordia Technologies
                    331 Newman Springs Road
                    Red Bank, NJ 07701
                    USA
           Tel:     +1 732 758 5254
           E-mail:  kaj@research.telcordia.com"
       DESCRIPTION
         "Copyright (C) The Internet Society (2003). This version of
         this MIB module is part of RFC 3606; see the RFC itself for
         full legal notices.

         This MIB Module is a supplement to the ATM-MIB
         defined in RFC 2515."
       REVISION "200309230000Z"
       DESCRIPTION
         "Initial version of this MIB, published as RFC 3606."

         ::= { atmMIBObjects 14 }

   atm2MIBObjects  OBJECT IDENTIFIER ::= {atm2MIB 1}

   atm2MIBTraps OBJECT IDENTIFIER ::= {atm2MIB 2}

   -- This ATM2-MIB Module consists of the following tables,
   -- plus ATM trap support:
   --       1. atmSvcVpCrossConnectTable
   --       2. atmSvcVcCrossConnectTable
   --       3. atmSigStatTable
   --       4. atmSigSupportTable
   --       5. atmSigDescrParamTable
   --       6. atmIfRegisteredAddrTable
   --       7. atmVclAddrTable
   --       8. atmAddrVclTable
   --       9. atmVplStatTable
   --       10. atmVplLogicalPortTable
   --       11. atmVclStatTable
   --       12. atmAal5VclStatTable
   --       13. atmVclGenTable
   --       14. atmInterfaceExtTable
   --       15. atmIlmiSrvcRegTable
   --       16. atmIlmiNetworkPrefixTable
   --       17. atmSwitchAddressTable
   --       18. atmVpCrossConnectXTable
   --       19. atmVcCrossConnectXTable
   --       20. atmCurrentlyFailingPVplTable
   --       21. atmCurrentlyFailingPVclTable

   -- 1. ATM VPL SVC Cross-Connect Table

   atmSvcVpCrossConnectTable OBJECT-TYPE
```

```
       SYNTAX       SEQUENCE OF
                    AtmSvcVpCrossConnectEntry
       MAX-ACCESS   not-accessible
       STATUS       current
       DESCRIPTION
         "The ATM SVPC Cross-Connect table.  A
         bi-directional VP cross-connect between two
         switched VPLs is modeled as one entry in this
         table.  A Soft PVPC cross-connect, between a
         soft permanent VPL and a switched VPL, is
         also modeled as one entry in this table."
       ::= { atm2MIBObjects 1 }

   atmSvcVpCrossConnectEntry OBJECT-TYPE
       SYNTAX       AtmSvcVpCrossConnectEntry
       MAX-ACCESS   not-accessible
       STATUS       current
       DESCRIPTION
         "An entry in the ATM SVPC Cross-Connect table.
         This entry is used to model a bi-directional
         ATM VP cross-connect between two VPLs."
       INDEX { atmSvcVpCrossConnectIndex,
               atmSvcVpCrossConnectLowIfIndex,
               atmSvcVpCrossConnectLowVpi,
               atmSvcVpCrossConnectHighIfIndex,
               atmSvcVpCrossConnectHighVpi }
       ::= { atmSvcVpCrossConnectTable 1 }

   AtmSvcVpCrossConnectEntry ::= SEQUENCE {
       atmSvcVpCrossConnectIndex            INTEGER,
       atmSvcVpCrossConnectLowIfIndex       InterfaceIndex,
       atmSvcVpCrossConnectLowVpi           AtmVpIdentifier,
       atmSvcVpCrossConnectHighIfIndex      InterfaceIndex,
       atmSvcVpCrossConnectHighVpi          AtmVpIdentifier,
       atmSvcVpCrossConnectCreationTime     TimeStamp,
       atmSvcVpCrossConnectRowStatus        RowStatus
                     }

   atmSvcVpCrossConnectIndex OBJECT-TYPE
       SYNTAX       INTEGER (1..2147483647)
       MAX-ACCESS   not-accessible
       STATUS       current
       DESCRIPTION
         "A unique value to identify this SVPC
         cross-connect.  For each VP associated
         with this cross-connect, the agent reports
         this cross-connect index value in the
         atmVplCrossConnectIdentifer attribute of the
         corresponding atmVplTable entries."
       ::= { atmSvcVpCrossConnectEntry 1 }

   atmSvcVpCrossConnectLowIfIndex OBJECT-TYPE
       SYNTAX       InterfaceIndex
       MAX-ACCESS   not-accessible
       STATUS       current
       DESCRIPTION
         "The value of this object is equal to the
         ifIndex value of the ATM interface port for this
         SVPC cross-connect.  The term low implies
         that this ATM interface has the numerically lower
         ifIndex value than the other ATM interface
         identified in the same atmSvcVpCrossConnectEntry."
       ::= { atmSvcVpCrossConnectEntry 2 }

   atmSvcVpCrossConnectLowVpi OBJECT-TYPE
       SYNTAX       AtmVpIdentifier
       MAX-ACCESS   not-accessible
       STATUS       current
       DESCRIPTION
         "The value of this object is equal to the VPI
         value associated with the SVPC cross-connect
         at the ATM interface that is identified by
         atmSvcVpCrossConnectLowIfIndex.  The VPI value
         cannot exceed the number supported by the
         atmInterfaceCurrentMaxSvpcVpi at the low ATM interface
         port."
       ::= { atmSvcVpCrossConnectEntry 3 }

   atmSvcVpCrossConnectHighIfIndex OBJECT-TYPE
       SYNTAX       InterfaceIndex
       MAX-ACCESS   not-accessible
       STATUS       current
       DESCRIPTION
        "The value of this object is equal to the
         ifIndex value of the ATM interface port for
         this SVC VP cross-connect.  The term high
         implies that this ATM interface has the
         numerically higher ifIndex value than the
         other ATM interface identified in the same
         atmSvcVpCrossConnectEntry."
       ::= { atmSvcVpCrossConnectEntry 4 }

   atmSvcVpCrossConnectHighVpi OBJECT-TYPE
       SYNTAX       AtmVpIdentifier
       MAX-ACCESS   not-accessible
       STATUS       current
       DESCRIPTION
        "The value of this object is equal to the VPI
         value associated with the SVPC cross-connect
         at the ATM interface that is identified by
         atmSvcVpCrossConnectHighIfIndex.  The VPI value
         cannot exceed the number supported by the
         atmInterfaceCurrentMaxSvpcVpi at the high ATM interface
         port."
       ::= { atmSvcVpCrossConnectEntry 5 }

   atmSvcVpCrossConnectCreationTime OBJECT-TYPE
       SYNTAX       TimeStamp
       MAX-ACCESS   read-only
       STATUS       current
       DESCRIPTION
        "The value of the sysUpTime object
         at the time this bi-directional SVPC
         cross-connect was created.  If the current
         state was entered prior to the last
         re-initialization of the agent, then this
         object contains a zero value."
       ::= { atmSvcVpCrossConnectEntry 6 }

   atmSvcVpCrossConnectRowStatus OBJECT-TYPE
       SYNTAX       RowStatus
       MAX-ACCESS   read-write
       STATUS       current
       DESCRIPTION
           "This object is used to delete rows in the
            atmSvcVpCrossConnectTable."
       ::= { atmSvcVpCrossConnectEntry 7 }

   -- 2. ATM VCL SVC Cross-Connect Table

   atmSvcVcCrossConnectTable OBJECT-TYPE
       SYNTAX       SEQUENCE OF AtmSvcVcCrossConnectEntry
       MAX-ACCESS   not-accessible
       STATUS       current
       DESCRIPTION
        "The ATM SVCC Cross-Connect table.  A
         bi-directional VC cross-connect between two
         switched VCLs is modeled as one entry in
         this table.  A Soft PVCC cross-connect,
         between a soft permanent VCL and a switched
         VCL, is also modeled as one entry in this
         table."
       ::= { atm2MIBObjects 2 }
   atmSvcVcCrossConnectEntry OBJECT-TYPE
       SYNTAX       AtmSvcVcCrossConnectEntry
       MAX-ACCESS   not-accessible
       STATUS       current
       DESCRIPTION
        "An entry in the ATM SVCC Cross-Connect table.
         This entry is used to model a bi-directional ATM
         VC cross-connect between two VCLs."
       INDEX { atmSvcVcCrossConnectIndex,
               atmSvcVcCrossConnectLowIfIndex,
               atmSvcVcCrossConnectLowVpi,
               atmSvcVcCrossConnectLowVci,
               atmSvcVcCrossConnectHighIfIndex,
               atmSvcVcCrossConnectHighVpi,
               atmSvcVcCrossConnectHighVci }
       ::= { atmSvcVcCrossConnectTable 1 }

   AtmSvcVcCrossConnectEntry ::= SEQUENCE {
       atmSvcVcCrossConnectIndex           INTEGER,
       atmSvcVcCrossConnectLowIfIndex      InterfaceIndex,
       atmSvcVcCrossConnectLowVpi          AtmVpIdentifier,
       atmSvcVcCrossConnectLowVci          AtmVcIdentifier,
       atmSvcVcCrossConnectHighIfIndex     InterfaceIndex,
       atmSvcVcCrossConnectHighVpi         AtmVpIdentifier,
       atmSvcVcCrossConnectHighVci         AtmVcIdentifier,
       atmSvcVcCrossConnectCreationTime    TimeStamp,
       atmSvcVcCrossConnectRowStatus       RowStatus
               }

   atmSvcVcCrossConnectIndex OBJECT-TYPE
       SYNTAX       INTEGER (1..2147483647)
       MAX-ACCESS   not-accessible
       STATUS       current
       DESCRIPTION
        "A unique value to identify this SVCC cross-connect.
         For each VP associated with this cross-connect, the
         agent reports this cross-connect index value in the
         atmVclCrossConnectIdentifier attribute of the
         corresponding atmVplTable entries."
       ::= { atmSvcVcCrossConnectEntry 1 }

   atmSvcVcCrossConnectLowIfIndex OBJECT-TYPE
       SYNTAX       InterfaceIndex
       MAX-ACCESS   not-accessible
       STATUS       current
       DESCRIPTION
        "The value of this object is equal to the
         ifIndex value of the ATM interface port for this
         SVCC cross-connect.  The term low implies that
         this ATM interface has the numerically lower
         ifIndex value than the other ATM interface
         identified in the same atmSvcVcCrossConnectEntry."
       ::= { atmSvcVcCrossConnectEntry 2 }

   atmSvcVcCrossConnectLowVpi OBJECT-TYPE
       SYNTAX       AtmVpIdentifier
       MAX-ACCESS   not-accessible
       STATUS       current
       DESCRIPTION
        "The value of this object is equal to the VPI
         value associated with the SVCC cross-connect
         at the ATM interface that is identified by
         atmSvcVcCrossConnectLowIfIndex.  The VPI value
         cannot exceed the number supported by the
         atmInterfaceCurrentMaxSvccVpi at the low ATM interface
         port."
       ::= { atmSvcVcCrossConnectEntry 3 }

   atmSvcVcCrossConnectLowVci OBJECT-TYPE
       SYNTAX       AtmVcIdentifier
       MAX-ACCESS   not-accessible
       STATUS       current
       DESCRIPTION
        "The value of this object is equal to the VCI
         value associated with the SVCC cross-connect
         at the ATM interface that is identified by
         atmSvcVcCrossConnectLowIfIndex.  The VCI value
         cannot exceed the number supported by the
         atmInterfaceCurrentMaxSvccVci at the low ATM interface
         port."
       ::= { atmSvcVcCrossConnectEntry 4 }

   atmSvcVcCrossConnectHighIfIndex OBJECT-TYPE
       SYNTAX       InterfaceIndex
       MAX-ACCESS   not-accessible
       STATUS       current
       DESCRIPTION
        "The value of this object is equal to the
         ifIndex value for the ATM interface port for
         this SVCC cross-connect.  The term high implies
         that this ATM interface has the numerically
         higher ifIndex value than the other ATM interface
         identified in the same atmSvcVcCrossConnectEntry."
       ::= { atmSvcVcCrossConnectEntry 5 }

   atmSvcVcCrossConnectHighVpi OBJECT-TYPE
```

```
       SYNTAX       AtmVpIdentifier
       MAX-ACCESS   not-accessible
       STATUS       current
       DESCRIPTION
        "The value of this object is equal to the VPI
         value associated with the SVCC cross-connect
         at the ATM interface that is identified by
         atmSvcVcCrossConnectHighIfIndex.  The VPI value
         cannot exceed the number supported by the
         atmInterfaceCurrentMaxSvccVpi at the high ATM interface
         port."
       ::= { atmSvcVcCrossConnectEntry 6 }

   atmSvcVcCrossConnectHighVci OBJECT-TYPE
       SYNTAX       AtmVcIdentifier
       MAX-ACCESS   not-accessible
       STATUS       current
       DESCRIPTION
        "The value of this object is equal to the VCI
         value associated with the SVCC cross-connect
         at the ATM interface that is identified by
         atmSvcVcCrossConnectHighIfIndex.  The VCI value
         cannot exceed the number supported by the
         atmInterfaceMaxVciBits at the high ATM interface
         port."
       ::= { atmSvcVcCrossConnectEntry 7 }

   atmSvcVcCrossConnectCreationTime OBJECT-TYPE
       SYNTAX       TimeStamp
       MAX-ACCESS   read-only
       STATUS       current
       DESCRIPTION
        "The value of the sysUpTime object
         at the time this bi-directional SVCC
         cross-connect was created.  If the current
         state was entered prior to the last
         re-initialization of the agent, then this
         object contains a zero value."
       ::= { atmSvcVcCrossConnectEntry 8 }

   atmSvcVcCrossConnectRowStatus OBJECT-TYPE
       SYNTAX       RowStatus
       MAX-ACCESS   read-write
       STATUS       current
       DESCRIPTION
           "This object is used to delete rows in the
            atmSvcVcCrossConnectTable."
       ::= { atmSvcVcCrossConnectEntry 9 }
   -- 3. ATM Interface Signalling Statistics Table --

   atmSigStatTable       OBJECT-TYPE
       SYNTAX       SEQUENCE OF AtmSigStatEntry
       MAX-ACCESS   not-accessible
       STATUS       current
       DESCRIPTION
        "This table contains ATM interface signalling
        statistics, one entry per ATM signalling
        interface."
       ::= { atm2MIBObjects 3 }

   atmSigStatEntry         OBJECT-TYPE
       SYNTAX       AtmSigStatEntry
       MAX-ACCESS   not-accessible
       STATUS       current
       DESCRIPTION
        "This list contains signalling statistics variables."
       INDEX { ifIndex }
       ::= { atmSigStatTable  1}

   AtmSigStatEntry    ::= SEQUENCE  {
       atmSigSSCOPConEvents            Counter32,
       atmSigSSCOPErrdPdus             Counter32,
       atmSigDetectSetupAttempts       Counter32,
       atmSigEmitSetupAttempts         Counter32,
       atmSigDetectUnavailRoutes       Counter32,
       atmSigEmitUnavailRoutes         Counter32,
       atmSigDetectUnavailResrcs       Counter32,
       atmSigEmitUnavailResrcs         Counter32,
       atmSigDetectCldPtyEvents        Counter32,
       atmSigEmitCldPtyEvents          Counter32,
       atmSigDetectMsgErrors           Counter32,
       atmSigEmitMsgErrors             Counter32,
       atmSigDetectClgPtyEvents        Counter32,
       atmSigEmitClgPtyEvents          Counter32,
       atmSigDetectTimerExpireds       Counter32,
       atmSigEmitTimerExpireds         Counter32,
       atmSigDetectRestarts            Counter32,
       atmSigEmitRestarts              Counter32,
       atmSigInEstabls                 Counter32,
       atmSigOutEstabls                Counter32
            }

   atmSigSSCOPConEvents   OBJECT-TYPE
       SYNTAX       Counter32
       MAX-ACCESS   read-only
       STATUS       current
       DESCRIPTION
           "SSCOP Connection Events Counter.  This counter counts
            the sum of the following errors:

            1) SSCOP Connection Disconnect Counter

                The abnormal occurrence of this event is
                characterized by the expiry of Timer_NO_RESPONSE.
                (This event is communicated to the layer management
                with MAA-ERROR code P.  See ITU-T Q.2110.)

             2) SSCOP Connection Initiation Failure

                This condition indicates the inability to establish
                an SSCOP connection.  This event occurs whenever the
                number of expiries of the connection control timer
                (Timer_CC) equals or exceeds the MaxCC, or upon
                receipt of a connection reject message BGREJ PDU.
                (This event is communicated to layer management with
                MAA-ERROR code O.  See ITU-T Q.2110.)

             3) SSCOP Connection Re-Establ/Resynch

                This event occurs upon receipt of a BGN PDU or
                RS PDU."
       REFERENCE
           "ITU-T Recommendation Q.2110, Broadband
            Integrated Services Digital Network
            (B-ISDN) - ATM Adaptation Layer - Service
            Specific Connection Oriented Protocol (SSCOP)
            Specification, July 1994."
       ::= { atmSigStatEntry 1}

   atmSigSSCOPErrdPdus   OBJECT-TYPE
       SYNTAX       Counter32
       MAX-ACCESS   read-only
       STATUS       current
       DESCRIPTION
           "SSCOP Errored PDUs Counter.  This counter counts the
           sum of the following errors:

           1) Invalid PDUs.
              These are defined in SSCOP and consist of PDUs
              with an incorrect length (MAA-ERROR code U), an
              undefined PDU type code, or that are not 32-bit
              aligned.

           2) PDUs that result in MAA-ERROR codes and are
```

```
               discarded.

            See MAA-ERROR codes A-D, F-M, and Q-T defined in
            ITU-T Q.2110."
        REFERENCE
            "ITU-T Recommendation Q.2110, Broadband
             Integrated Services Digital Network
             (B-ISDN) - ATM Adaptation Layer - Service
             Specific Connection Oriented Protocol (SSCOP)
             Specification, July 1994."
        ::= { atmSigStatEntry 2 }

    atmSigDetectSetupAttempts    OBJECT-TYPE
        SYNTAX       Counter32
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION
            "Call Setup Attempts Counter.  This counter counts
            the number of call setup attempts (both successful
            and unsuccessful) detected on this interface."
        ::= { atmSigStatEntry 3 }

    atmSigEmitSetupAttempts    OBJECT-TYPE
        SYNTAX       Counter32
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION
            "Call Setup Attempts Counter.  This counter counts
            the number of call setup attempts (both successful
            and unsuccessful) transmitted on this interface."
        ::= { atmSigStatEntry 4 }

    atmSigDetectUnavailRoutes    OBJECT-TYPE
        SYNTAX       Counter32
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION
            "Number of Route Unavailability detected on this interface.
            This counter is incremented when a RELEASE, RELEASE COMPLETE
            (only when not preceded by a RELEASE message for the same
            call), ADD PARTY REJECT, or STATUS message that
            contains one of the following cause code values is
            received (Note: These cause values
            apply to both UNI3.0 and UNI3.1):

            Cause Value            Meaning
```

```
              1            unallocated (unassigned) number
              2            no route to specified transit network
              3            no route to destination

           NOTE:  For this counter, RELEASE COMPLETE
           messages that are a reply to a previous RELEASE
           message and contain the same cause value, are
           redundant (for counting purposes) and should not
           be counted."

       ::= { atmSigStatEntry 5 }

   atmSigEmitUnavailRoutes    OBJECT-TYPE
       SYNTAX       Counter32
       MAX-ACCESS   read-only
       STATUS       current
       DESCRIPTION
           "Number of Route Unavailability transmitted from this
           interface.  This counter is incremented when a RELEASE,
           RELEASE COMPLETE (only when not preceded by a RELEASE
           message for the same call), ADD PARTY REJECT, or
           STATUS message that contains one of the following cause
           code values is transmitted (Note: These cause values apply
           to both UNI3.0 and UNI3.1):

           Cause Value            Meaning

              1            unallocated (unassigned) number
              2            no route to specified transit network
              3            no route to destination

           NOTE:  For this counter, RELEASE COMPLETE
           messages that are a reply to a previous RELEASE
           message and contain the same cause value, are
           redundant (for counting purposes) and should not
           be counted."
       ::= { atmSigStatEntry 6 }

   atmSigDetectUnavailResrcs     OBJECT-TYPE
       SYNTAX       Counter32
       MAX-ACCESS   read-only
       STATUS       current
       DESCRIPTION
           "Number of Resource Unavailability detected on this
           interface.  This counter is incremented when a RELEASE,
           RELEASE COMPLETE (only when not preceded by a RELEASE
           message for the same call), ADD PARTY REJECT, or
           STATUS message that contains one of the following
           cause code values is received (Note: These cause
           values apply to both UNI3.0 and UNI3.1 unless
           otherwise stated):

           Cause Value          Meaning

              35          requested VPCI/VCI not available
              37          user cell rate not available (UNI3.1
                          only)
              38          network out of order
              41          temporary failure
              45          no VPCI/VCI available
              47          resource unavailable, unspecified
              49          Quality of Service unavailable
              51          user cell rate not available (UNI3.0
                          only)
              58          bearer capability not presently
                          available
              63          Service or option not available,
                          unspecified
              92          too many pending add party requests

           NOTE:  For this counter, RELEASE COMPLETE
           messages that are a reply to a previous RELEASE
           message and contain the same cause value, are
           redundant (for counting purposes) and should not
           be counted."
       ::= { atmSigStatEntry 7 }

   atmSigEmitUnavailResrcs     OBJECT-TYPE
       SYNTAX       Counter32
       MAX-ACCESS   read-only
       STATUS       current
       DESCRIPTION
          "Number of Resource Unavailability transmitted from this
          interface.  This counter is incremented when a RELEASE,
          RELEASE COMPLETE (only when not preceded by a RELEASE message
          for the same call), ADD PARTY REJECT, or STATUS message that
          contains one of the following cause code values is transmitted
          (Note: These cause values apply to both UNI3.0 and UNI3.1
          unless otherwise stated):

          Cause Value          Meaning

             35          requested VPCI/VCI not available
             37          user cell rate not available (UNI3.1
                         only)
             38          network out of order
```

```
                 41          temporary failure
                 45          no VPCI/VCI available
                 47          resource unavailable, unspecified
                 49          Quality of Service unavailable
                 51          user cell rate not available (UNI3.0
                             only)
                 58          bearer capability not presently
                             available
                 63          Service or option not available,
                             unspecified
                 92          too many pending add party requests

               NOTE:  For this counter, RELEASE COMPLETE messages that are a
               reply to a previous RELEASE message and contain the same cause
               value, are redundant (for counting purposes) and should not be
               counted."

           ::= { atmSigStatEntry 8 }

       atmSigDetectCldPtyEvents     OBJECT-TYPE
           SYNTAX       Counter32
           MAX-ACCESS   read-only
           STATUS       current
           DESCRIPTION
              "Number of Called Party Responsible For Unsuccessful Call
              detected on this interface.  This counter is incremented when a
              RELEASE, RELEASE COMPLETE (only when not preceded by a RELEASE
              message for the same call), ADD PARTY REJECT, or STATUS message
              that contains one of the following cause code values is
              received (Note: These cause values apply to both UNI3.0 and
              UNI3.1):

              Cause Value         Meaning

                 17          user busy
                 18          no user responding
                 21          call rejected
                 22          number changed
                 23          user rejects all calls with calling
                             line identification restriction (CLIR)
                 27          destination out of order
                 31          normal, unspecified
                 88          incompatible destination


              NOTE:  For this counter, RELEASE COMPLETE messages that are a
              reply to a previous RELEASE message and contain the same cause
              value, are redundant (for counting purposes) and should not be
```

```
         counted.

         Note: Cause Value #30 'response to STATUS ENQUIRY' was not
         included in this memo since it did not apply to a hard
         failure."

     ::= { atmSigStatEntry 9 }

 atmSigEmitCldPtyEvents     OBJECT-TYPE
     SYNTAX       Counter32
     MAX-ACCESS   read-only
     STATUS       current
     DESCRIPTION

        "Number of Called Party Responsible For Unsuccessful Call
        transmitted from this interface.  This counter is incremented
        when a RELEASE, RELEASE COMPLETE (only when not preceded by a
        RELEASE message for the same call), ADD PARTY REJECT, or STATUS
        message that contains one of the following cause code values is
        transmitted (Note: These cause values apply to both UNI3.0 and
        UNI3.1):

        Cause Value         Meaning

           17           user busy
           18           no user responding
           21           call rejected
           22           number changed
           23           user rejects all calls with calling
                        line identification restriction (CLIR)
           27           destination out of order
           31           normal, unspecified
           88           incompatible destination

        NOTE:  For this counter, RELEASE COMPLETE messages that are a
        reply to a previous RELEASE message and contain the same cause
        value, are redundant (for counting purposes) and should not be
        counted.

        Note: Cause Value #30 'response to STATUS ENQUIRY' was not
        included in this memo since it did not apply to a hard failure."

    ::= { atmSigStatEntry 10 }

atmSigDetectMsgErrors    OBJECT-TYPE
    SYNTAX       Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "Number of Incorrect Messages detected on this interface.  The
       Incorrect Messages Counter reflects any sort of incorrect
       information in a message.  This includes:

       - RELEASE, RELEASE COMPLETE, ADD PARTY REJECT,
         and STATUS messages transmitted, that contain any of
         the Cause values listed below.

       - Ignored messages.  These messages are dropped
         because the message was so damaged that it could
         not be further processed.  A list of dropped
         messages is compiled below:

          1.  Message with invalid protocol discriminator

          2.  Message with errors in the call reference I.E.
               - Bits 5-8 of the first octet not equal to
                 '0000'
               - Bits 1-4 of the first octet indicating a
                 length other than 3 octets
               - RELEASE COMPLETE message received with a
                 call reference that does not relate to a
                 call active or in progress
               - SETUP message received with call reference
                 flag incorrectly set to 1
               - SETUP message received with a call
                 reference for a call that is already
                 active or in progress.

          3.  Message too short

       The following cause values are monitored by this counter (Note:
       These cause values apply to both UNI3.0 and UNI3.1 unless
       otherwise stated):

       Cause Value     Meaning

          10     VPCI/VCI unacceptable (UNI3.0 only)
          36     VPCI/VCI assignment failure (UNI3.1 only)
          81     invalid call reference value
          82     identified channel does not exist
          89     invalid endpoint reference
          96     mandatory information element is missing
          97     message type non-existent or not
                 implemented
          99     information element non-existent or not
                 implemented
          100    invalid information element contents
          101    message not compatible with call state
          104    incorrect message length
          111    protocol error, unspecified

          NOTE:  For this counter, RELEASE COMPLETE messages that are
          a reply to a previous RELEASE message and contain the same
          cause value, are redundant (for counting purposes) and
          should not be counted."

    ::= { atmSigStatEntry 11 }

atmSigEmitMsgErrors    OBJECT-TYPE
    SYNTAX       Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "Number of Incorrect Messages transmitted on this interface.
       The Incorrect Messages Counter reflects any sort of incorrect
       information in a message.  This includes:

       - RELEASE, RELEASE COMPLETE, ADD PARTY REJECT,
         and STATUS messages transmitted or
         received, that contain any of the Cause values
         listed below.

       - Ignored messages.  These messages are dropped
         because the message was so damaged that it could
         not be further processed.  A list of dropped
         messages is compiled below:

          1.  Message with invalid protocol discriminator

          2.  Message with errors in the call reference I.E.
                - Bits 5-8 of the first octet not equal to
                  '0000'
                - Bits 1-4 of the first octet indicating a
                  length other than 3 octets
                - RELEASE COMPLETE message received with a
                  call reference that does not relate to a
                  call active or in progress
                - SETUP message received with call reference
                  flag incorrectly set to 1
                - SETUP message received with a call
                  reference for a call that is already
                  active or in progress.

          3.  Message too short
       The following cause values are monitored by this counter
       (Note: These cause values apply to both UNI3.0 and UNI3.1
       unless otherwise stated):

       Cause Value     Meaning

          10     VPCI/VCI unacceptable (UNI3.0 only)
          36     VPCI/VCI assignment failure (UNI3.1 only)
          81     invalid call reference value
          82     identified channel does not exist
          89     invalid endpoint reference
          96     mandatory information element is missing
          97     message type non-existent or not
                 implemented
          99     information element non-existent or not
                 implemented
          100    invalid information element contents
          101    message not compatible with call state
          104    incorrect message length
          111    protocol error, unspecified

          NOTE:  For this counter, RELEASE COMPLETE messages that are
          a reply to a previous RELEASE message and contain the same
          cause value, are redundant (for counting purposes) and
          should not be counted."

    ::= { atmSigStatEntry 12 }

atmSigDetectClgPtyEvents    OBJECT-TYPE
    SYNTAX       Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "Number of Calling Party Events detected on this interface.
       This counter monitors error events that occur due to the
       originating user doing something wrong.  This counter is
       incremented when a RELEASE, RELEASE COMPLETE (only when not
       preceded by a RELEASE message for the same call), ADD PARTY
       REJECT, or STATUS message that contains one of the following
       cause code values is received (Note: These cause values
       apply to both UNI3.0 and UNI3.1):

       Cause Value     Meaning

          28     invalid number format (address incomplete)
          43     access information discarded
          57     bearer capability not authorized
          65     bearer capability not implemented
          73     unsupported combination of traffic
                 parameters
          78     AAL parameters cannot be supported (UNI3.1
                 only)
          91     invalid transit network selection
          93     AAL parameters cannot be supported (UNI3.0
                 only)

          NOTE:  For this counter, RELEASE COMPLETE messages that
          are a reply to a previous RELEASE message and contain
          the same cause value, are redundant (for counting purposes)
          and should not be counted."

    ::= { atmSigStatEntry 13 }

atmSigEmitClgPtyEvents    OBJECT-TYPE
    SYNTAX       Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "Number of Calling Party Events transmitted from this interface.
       This counter monitors error events that occur due to the
       originating user doing something wrong.  This counter is
       incremented when a RELEASE, RELEASE COMPLETE (only when not
       preceded by a RELEASE message for the same call), ADD PARTY
       REJECT, or STATUS message that contains one of the following
       cause code values is transmitted (Note: These cause values
       apply to both UNI3.0 and UNI3.1):

       Cause Value     Meaning

          28     invalid number format (address incomplete)
          43     access information discarded
          57     bearer capability not authorized
          65     bearer capability not implemented
          73     unsupported combination of traffic
                 parameters
          78     AAL parameters cannot be supported (UNI3.1
                 only)
          91     invalid transit network selection
          93     AAL parameters cannot be supported (UNI3.0
                 only)

          NOTE:  For this counter, RELEASE COMPLETE messages that are
          a reply to a previous RELEASE message and contain the same
          cause value, are redundant (for counting purposes) and
          should not be counted."
    ::= { atmSigStatEntry 14 }

atmSigDetectTimerExpireds    OBJECT-TYPE
    SYNTAX       Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "Number of Timer Expiries detected on this interface.  The Timer
       Expiries Counter provides a count of network timer expiries, and
       to some extent, host or switch timer expiries.  The conditions
       for incrementing this counter are:

          - Expiry of any network timer

          - Receipt of a RELEASE or RELEASE COMPLETE
            message with Cause #102, ’recovery on
            timer expiry’.

          NOTE:  For this counter, RELEASE COMPLETE messages that are
          a reply to a previous RELEASE message and contain the same
          cause value, are redundant (for counting purposes) and
          should not be counted."

    ::= { atmSigStatEntry 15 }

atmSigEmitTimerExpireds    OBJECT-TYPE
    SYNTAX       Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "Number of Timer Expiries transmitted from this interface.
       The Timer Expiries Counter provides a count of network timer
       expiries, and to some extent, host or switch timer expiries.
       The conditions for incrementing this counter are:

          - Expiry of any network timer

          - Receipt of a RELEASE or RELEASE COMPLETE
            message with Cause #102, ’recovery on
            timer expiry’.

       NOTE:  For this counter, RELEASE COMPLETE messages that are a
       reply to a previous RELEASE message and contain the same cause
       value, are redundant (for counting purposes) and should not be
       counted."

    ::= { atmSigStatEntry 16 }
```

```
atmSigDetectRestarts    OBJECT-TYPE
    SYNTAX       Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "Number of Restart Activity errors detected on this interface.
       The Restart Activity Counter provides a count of host, switch,
       or network restart activity.  This counter is incremented when
       receiving a RESTART message."

    ::= { atmSigStatEntry 17 }

atmSigEmitRestarts    OBJECT-TYPE
    SYNTAX       Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "Number of Restart Activity errors transmitted from this
       interface.  The Restart Activity Counter provides a count of
       host, switch, or network restart activity.  This counter is
       incremented when transmitting a RESTART message."

    ::= { atmSigStatEntry 18 }

atmSigInEstabls    OBJECT-TYPE
    SYNTAX       Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "Number of SVCs established at this signalling entity for
       incoming connections."
    ::= { atmSigStatEntry 19 }

atmSigOutEstabls    OBJECT-TYPE
    SYNTAX       Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "Number of SVCs established at this signalling entity for
       outgoing connections."
    ::= { atmSigStatEntry 20 }

  -- 4. ATM Interface Signalling Support Table
  --
  -- This table provides information to support
  -- the signalling process which is used to establish
  -- ATM Switched Virtual Connections (SVCs).
```

```
atmSigSupportTable       OBJECT-TYPE
    SYNTAX       SEQUENCE OF AtmSigSupportEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "This table contains ATM local interface configuration
       parameters, one entry per ATM signalling interface."
    ::= { atm2MIBObjects 4 }

atmSigSupportEntry         OBJECT-TYPE
    SYNTAX       AtmSigSupportEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "This list contains signalling configuration parameters
       and state variables."
    INDEX { ifIndex }
    ::= { atmSigSupportTable  1}

AtmSigSupportEntry    ::= SEQUENCE  {
    atmSigSupportClgPtyNumDel       INTEGER,
    atmSigSupportClgPtySubAddr      INTEGER,
    atmSigSupportCldPtySubAddr      INTEGER,
    atmSigSupportHiLyrInfo          INTEGER,
    atmSigSupportLoLyrInfo          INTEGER,
    atmSigSupportBlliRepeatInd      INTEGER,
    atmSigSupportAALInfo            INTEGER,
    atmSigSupportPrefCarrier        OCTET STRING
}

atmSigSupportClgPtyNumDel     OBJECT-TYPE
    SYNTAX       INTEGER  { enabled(1), disabled(2) }
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
       "This object indicates whether the Calling Party Number
       Information Element is transferred to the called party
       address.  The value of this object can be:

         - enabled(1)  This Information Element is transferred
                       to the called party

         - disabled(2) This Information Element is NOT
                       transferred to the called party."

    ::= { atmSigSupportEntry 1 }

atmSigSupportClgPtySubAddr   OBJECT-TYPE
```

```
    SYNTAX       INTEGER  { enabled(1), disabled(2) }
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
       "This object indicates whether to accept and transfer the Calling
       Party Subaddress Information Element from the calling party to
       the called party.  Calling party subaddress information shall
       only be transferred to the called party if calling party number
       delivery is enabled (i.e., atmSigSupportClgPtyNumDel =
       'enabled(1)'.  The value of this object can be:

         - enabled(1)  This Information Element is transferred
                       to the called party

         - disabled(2) This Information Element is NOT
                       transferred to the called party."

    ::= { atmSigSupportEntry 2 }

atmSigSupportCldPtySubAddr    OBJECT-TYPE
    SYNTAX       INTEGER  { enabled(1), disabled(2) }
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
       "This object indicates whether to accept, transfer, and deliver
       the Called Party Subaddress Information Element from the calling
       party to the called party.  The value of this object can be:

         - enabled(1)  This Information Element is transferred
                       to the called party

         - disabled(2) This Information Element is NOT
                       transferred to the called party."

    ::= { atmSigSupportEntry 3 }

atmSigSupportHiLyrInfo     OBJECT-TYPE
    SYNTAX       INTEGER  { enabled(1), disabled(2) }
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
       "This object indicates whether to accept, transfer, and deliver
       the Broadband High Layer Information Element from the calling
       party to the called party.  The value of this object can be:

         - enabled(1)  This Information Element is transferred
                       to the called party
         - disabled(2) This Information Element is NOT
                       transferred to the called party."

    ::= { atmSigSupportEntry 4 }

atmSigSupportLoLyrInfo     OBJECT-TYPE
    SYNTAX       INTEGER  { enabled(1), disabled(2) }
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
       "This object indicates whether to accept, transfer, and deliver
       the Broadband Low Layer Information Element from the calling
       party to the called party.  The value of this object can be:

         - enabled(1)  This Information Element is transferred
                       to the called party

         - disabled(2) This Information Element is NOT
                       transferred to the called party."

    ::= { atmSigSupportEntry 5 }

atmSigSupportBlliRepeatInd      OBJECT-TYPE
    SYNTAX       INTEGER  { enabled(1), disabled(2) }
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "This object indicates whether to accept, transfer, and deliver
        the Broadband Repeat Indicator with two or three instances of
        the Broadband Low Layer Information Element for low layer
        information selection from the calling party to the called
        party.  This object’s value should always be disabled(2) if
        the value of atmSigSupportLolyrInfo is disabled(2).

        The value of this object can be:

        - enabled(1)  This Information Element is transferred
              to the called party

        - disabled(2) This Information Element is NOT
              transferred to the called party."
    ::= { atmSigSupportEntry 6 }

atmSigSupportAALInfo     OBJECT-TYPE
    SYNTAX       INTEGER  { enabled(1), disabled(2) }
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
```

```
           "This object indicates whether to accept, transfer, and deliver
           the ATM Adaptation Layer Parameters Information Element from the
           calling party to the called party.  The value of this object can
           be:

             - enabled(1)  This Information Element is transferred
                           to the called party

             - disabled(2) This Information Element is NOT
                           transferred to the called party."

        ::= { atmSigSupportEntry 7 }

atmSigSupportPrefCarrier     OBJECT-TYPE
        SYNTAX       OCTET STRING (SIZE(0|4))
        MAX-ACCESS   read-write
        STATUS       current
        DESCRIPTION
           "This parameter identifies the carrier to which intercarrier
           calls originated from this interface are routed when transit
           network selection information is not provided by the calling
           party.  If a Carrier Identification Code (CIC) is used the
           parameter shall contain the CIC.  For three-digit CICs, the first
           octet shall be '0' and the CIC is contained in the three
           following octets.  If the preferred carrier feature is not
           supported the value is a zero-length string."

        ::= { atmSigSupportEntry 8 }

  -- 5. ATM Signalling Descriptor Parameter Table

        atmSigDescrParamTable     OBJECT-TYPE
            SYNTAX       SEQUENCE OF AtmSigDescrParamEntry
            MAX-ACCESS   not-accessible
            STATUS       current
            DESCRIPTION
                "A table contains signalling capabilities of VCLs except the
                Traffic Descriptor.  Traffic descriptors are described in
                the atmTrafficDescrParamTable."

            REFERENCE
                "ATM User-Network Interface Specification, Version 3.1 (UNI
                3.1), September 1994, Section 5.4.5 Variable Length
                Information Elements."

            ::= { atm2MIBObjects 5 }

        atmSigDescrParamEntry        OBJECT-TYPE
            SYNTAX       AtmSigDescrParamEntry
            MAX-ACCESS   not-accessible
            STATUS       current
            DESCRIPTION
                "Each entry in this table represents a
                set of signalling capabilities that can
                be applied to a VCL.  There is no requirement
                for unique entries, except that the index must
                be unique."
            INDEX { atmSigDescrParamIndex }
            ::= { atmSigDescrParamTable 1 }

        AtmSigDescrParamEntry ::=
            SEQUENCE {
                atmSigDescrParamIndex
                                       AtmSigDescrParamIndex,
                atmSigDescrParamAalType         INTEGER,
                atmSigDescrParamAalSscsType     INTEGER,
                atmSigDescrParamBhliType        INTEGER,


                atmSigDescrParamBhliInfo        OCTET STRING,
                atmSigDescrParamBbcConnConf     INTEGER,
                atmSigDescrParamBlliLayer2      INTEGER,
                atmSigDescrParamBlliLayer3      INTEGER,
                atmSigDescrParamBlliPktSize     INTEGER,
                atmSigDescrParamBlliSnapId      INTEGER,
                atmSigDescrParamBlliOuiPid      OCTET STRING,
                atmSigDescrParamRowStatus       RowStatus
            }

        atmSigDescrParamIndex OBJECT-TYPE
            SYNTAX       AtmSigDescrParamIndex
            MAX-ACCESS   not-accessible
            STATUS       current
            DESCRIPTION
                "The value of this object is used by the
                atmVclGenSigDescrIndex object in the atmVclGenTable to
                identify a row in this table."

            ::= { atmSigDescrParamEntry 1 }

         atmSigDescrParamAalType OBJECT-TYPE
             SYNTAX       INTEGER {
                other(1),       -- not defined
                aal1(2),        -- AAL type 1
                aal34(3),       -- AAL type 3/4
                aal5(4),        -- AAL type 5
                userDefined(5), -- User-Defined AAL
                aal2(6)         -- AAL type 2
                }
             MAX-ACCESS   read-create
             STATUS       current
             DESCRIPTION
                "The AAL type.  The value of this object is set to other(1)
                when not defined."

             DEFVAL { other }
             ::= { atmSigDescrParamEntry 2 }

         atmSigDescrParamAalSscsType OBJECT-TYPE
             SYNTAX       INTEGER {
                other(1),       -- other, or not used
                assured(2),     -- Data SSCS based on SSCOP
                                -- assured operation
                nonassured(3),  -- Data SSCS based on SSCOP
                                -- non-assured operation
                frameRelay(4),  -- frame relay SSCS
                null(5)         -- null

                }
             MAX-ACCESS   read-create
             STATUS       current
             DESCRIPTION
                "The SSCS type used by this entry."

             DEFVAL { other }
             ::= { atmSigDescrParamEntry 3 }

         atmSigDescrParamBhliType OBJECT-TYPE
             SYNTAX       INTEGER {
                other(1),         -- not defined
                iso(2),           -- ISO
                user(3),          -- User specific
                hiProfile(4),     -- Higher layer profile
                                  -- this enum applicable to
                                  -- UNI 3.0 only
                vendorSpecific(5) -- Vender specific
                                  -- application identifier
                }
             MAX-ACCESS   read-create
             STATUS       current
             DESCRIPTION
                "The Broadband high layer type."

             DEFVAL { other }
             ::= { atmSigDescrParamEntry 4 }

         atmSigDescrParamBhliInfo OBJECT-TYPE
             SYNTAX     OCTET STRING (SIZE(0..8))
             MAX-ACCESS read-create
             STATUS     current
             DESCRIPTION
                "The Broadband high layer information.  When
                atmSigDescrParamBhliType is set to iso(2), the value of this
                object is a zero length string.  When
                atmSigDescrParamBhliType is set to user(3), the value of
                this object is an octet string with length ranging from 0 to
                8.  When atmSigDescrParamBhliType is set to hiProfile(4),
                the value of this object is a length of 4 octet string
                containing user to user profile identifier.  When
                atmSigDescrParamBhliType is set to vendorSpecific(5), the
                value of this object is a length of 7 octet string, where
                the most significant 3 octets consist of a globally-
                administered OUI, and the least significant 4 octets are the
                vender administered application OUI."

             DEFVAL { ''H }

             ::= { atmSigDescrParamEntry 5 }

         atmSigDescrParamBbcConnConf OBJECT-TYPE
             SYNTAX       INTEGER {
                 ptp(1),   -- point-to-point
                 ptmp(2)   -- point-to-multipoint
                 }
             MAX-ACCESS   read-create
             STATUS       current
             DESCRIPTION
                 "The Broadband bearer capability user plane connection
                  configuration parameter."
             DEFVAL { ptp }
             ::= { atmSigDescrParamEntry 6 }

          atmSigDescrParamBlliLayer2 OBJECT-TYPE
              SYNTAX       INTEGER {
                 other(1),        -- not specified
                 iso1745(2),      -- Basic mode ISO 1745
                 q921(3),         -- CCITT Recommendation Q.921
                 x25linklayer(4), -- CCITT Recommendation X.25
                                  -- Link Layer
                 x25multilink(5), -- CCITT Recommendation X.25
                                  -- Multilink
                 lapb(6),         -- Extended LAPB; for half
```

```
                              -- duplex operation
             hdlcArm(7),      -- HDLC ARM (ISO 4335)
             hdlcNrm(8),      -- HDLC NRM (ISO 4335)
             hdlcAbm(9),      -- HDLC ABM (ISO 4335)
             iso88022(10),    -- LAN logical link control
                              -- (ISO 8802/2)
             x75slp(11),      -- CCITT Recommendation X.75,
                              -- single link
                              -- procedure (SLP)
             q922(12),        -- CCITT Recommendation Q.922
             userDef(13),     -- User specified
             iso7776(14)      -- ISO 7776 DTE-DTE operation
             }
          MAX-ACCESS   read-create
          STATUS       current
          DESCRIPTION
            "The Broadband low layer information, protocol type of layer
            2.  The value of this object is other(1) if layer 2 protocol
            is not used."

          DEFVAL { other }
          ::= { atmSigDescrParamEntry 7 }

      atmSigDescrParamBlliLayer3 OBJECT-TYPE
          SYNTAX       INTEGER {
             other(1),       -- not specified
             x25pkt(2),      -- CCITT Recommendation X.25
                             -- packet layer
             isoiec8208(3),  -- ISO/IEC 8208 (X.25 packet
                             -- level protocol for data
                             -- terminal equipment)
             x223iso8878(4), -- X.223/ISO 8878
             isoiec8473(5),  -- ISO/IEC 8473 OSI
                             -- connectionless
                             -- mode protocol
             t70(6),         -- CCITT Recommendation T.70
                             -- minimum
                             -- network layer
             tr9577(7),      -- ISO/IEC TR 9577 Protocol
                             -- Identification in the
                             -- network layer
             userDef(8)      -- user specified

             }
          MAX-ACCESS   read-create
          STATUS       current
          DESCRIPTION
            "The Broadband low layer information, protocol type of layer
            3.  The value of this object is other(1) if layer 3 protocol
            is not used."

          DEFVAL { other }
          ::= { atmSigDescrParamEntry 8 }

      atmSigDescrParamBlliPktSize OBJECT-TYPE
          SYNTAX       INTEGER {
             other(1),        -- not used
             s16(2),          -- 16 octets
             s32(3),          -- 32 octets
             s64(4),          -- 64 octets
             s128(5),         -- 128 octets
             s256(6),         -- 256 octets
             s512(7),         -- 512 octets
             s1024(8),        -- 1028 octets
             s2048(9),        -- 2048 octets
             s4096(10)        -- 4096 octets
          }
          MAX-ACCESS   read-create
          STATUS       current
          DESCRIPTION
            "The default packet size defined in B-LLI."

          DEFVAL { other }
          ::= { atmSigDescrParamEntry 9 }

      atmSigDescrParamBlliSnapId OBJECT-TYPE
          SYNTAX       INTEGER {
             other(1),  -- not used
             true(2),   -- SNAP ID is 1
             false(3)   -- SNAP ID is 0
          }
          MAX-ACCESS   read-create
          STATUS       current
          DESCRIPTION
            "The SNAP ID used for Broadband low layer protocol layer 3.
            The value of this object is other(1) if
            atmSigDescrParamBlliLayer3 is set to other(1)."

          DEFVAL { other }
          ::= { atmSigDescrParamEntry 10 }

      atmSigDescrParamBlliOuiPid OBJECT-TYPE
          SYNTAX       OCTET STRING (SIZE(0|5))
          MAX-ACCESS   read-create
          STATUS       current
          DESCRIPTION
```

```
            "The OUI/PID encoding for Broadband low layer protocol layer
            3.  The value of this object is a zero length string if
            atmSigDescrParamBlliLayer3 is set to other(1).  When used,
            it is always 5 octets with the most significant octet as the
            OUI Octet 1 and the least significant octet as the PID Octet
            2."

          DEFVAL { ’’H }
          ::= { atmSigDescrParamEntry 11 }

       atmSigDescrParamRowStatus OBJECT-TYPE
           SYNTAX       RowStatus
           MAX-ACCESS   read-create
           STATUS       current
           DESCRIPTION
            "This object is used to create and delete rows in the
            atmSigDescrParamTable."

           ::= { atmSigDescrParamEntry 12 }

  -- 6. ATM Interface Registered Address Table --

atmIfRegisteredAddrTable    OBJECT-TYPE
    SYNTAX       SEQUENCE OF AtmIfRegisteredAddrEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "This table contains a list of ATM addresses that can be used for
       calls to and from a given interface by a switch or service.  The
       ATM addresses are either registered by the endsystem via ILMI or
       statically configured.  This table does not expose PNNI
       reachability information.  ILMI registered addresses cannot be
       deleted using this table.  This table only applies to switches
       and network services."
    ::= { atm2MIBObjects 6 }

atmIfRegisteredAddrEntry    OBJECT-TYPE
    SYNTAX       AtmIfRegisteredAddrEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "An entry in the ATM Interface Registered Address table."

    INDEX { ifIndex, atmIfRegAddrAddress }
    ::= { atmIfRegisteredAddrTable  1}

AtmIfRegisteredAddrEntry    ::= SEQUENCE  {
    atmIfRegAddrAddress             AtmAddr,
    atmIfRegAddrAddressSource       INTEGER,
    atmIfRegAddrOrgScope            INTEGER,
    atmIfRegAddrRowStatus           RowStatus
         }

atmIfRegAddrAddress   OBJECT-TYPE
    SYNTAX       AtmAddr
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "An address registered for a given switch or service interface."

    ::= { atmIfRegisteredAddrEntry 1}

atmIfRegAddrAddressSource  OBJECT-TYPE
    SYNTAX       INTEGER  {
       other(1),
       static(2),
       dynamic(3)
       }
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "The type of address source for a given ATM Address. The value
       dynamic(3) is indicated when ILMI is used."
    ::= { atmIfRegisteredAddrEntry 2}

atmIfRegAddrOrgScope  OBJECT-TYPE
    SYNTAX       INTEGER {
        localNetwork(1),
        localNetworkPlusOne(2),
        localNetworkPlusTwo(3),
        siteMinusOne(4),
        intraSite(5),
        sitePlusOne(6),
        organizationMinusOne(7),
        intraOrganization(8),
        organizationPlusOne(9),
        communityMinusOne(10),
        intraCommunity(11),
        communityPlusOne(12),
        regional(13),
        interRegional(14),
        global(15)
    }
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
```

```
        "This object indicates the organizational scope for
        the referenced address.  The information of the
        referenced address shall not be distributed outside
        the indicated scope.   Refer to Annex 5.3 of ATM
        Forum UNI Signalling 4.0 for guidelines regarding
        the use of organizational scopes.

        This value cannot be configured for ILMI-registered
        addresses.

        The default values for organizational scope are
        localNetwork(1) for ATM group addresses, and
        global(15) for individual addresses."
    ::= { atmIfRegisteredAddrEntry 3}

atmIfRegAddrRowStatus OBJECT-TYPE
    SYNTAX       RowStatus
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
       "This object is used to create and delete rows in the
       atmIfRegisteredAddrTable.  Rows created dynamically (e.g., ILMI-
       registered addresses) cannot be deleted using this object."

    ::= { atmIfRegisteredAddrEntry 4}

  -- 7. ATM VPI/VCI to Address Mapping Table

    atmVclAddrTable  OBJECT-TYPE
        SYNTAX       SEQUENCE OF AtmVclAddrEntry
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION
            "This table provides a mapping between the atmVclTable and
            the ATM called party/calling party address.  This table can
            be used to retrieve the calling party and called party ATM
            address pair for a given VCL.  Note that there can be more
            than one pair of calling party and called party ATM
            addresses for a VCL in a point to multi-point call."
        ::= { atm2MIBObjects 7 }

    atmVclAddrEntry    OBJECT-TYPE
        SYNTAX       AtmVclAddrEntry
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION
            "Each entry in this table represents a binding between a VCL
            and an ATM address associated with this call.  This ATM
            address can be either the called party address or the
            calling party address.  There can be more than one pair of
            calling/called party ATM addresses associated with the VCL
            entry for point to multi-point calls.  Objects
            atmVclAddrType, and atmVclAddrRowStatus are
            required during row creation."
        INDEX  { ifIndex, atmVclVpi, atmVclVci,
                 atmVclAddrAddr }
        ::= { atmVclAddrTable 1 }

    AtmVclAddrEntry ::=
        SEQUENCE {
            atmVclAddrAddr          AtmAddr,
            atmVclAddrType          INTEGER,
            atmVclAddrRowStatus     RowStatus
        }

    atmVclAddrAddr     OBJECT-TYPE
        SYNTAX       AtmAddr
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION
            "An ATM address on one end of the VCL.  For SVCs, the agent
            supplies the value of this object at creation time.  For PVC
            VCL, the manager can supply the value of this object during
            or after the PVC VCL creation."
        ::= { atmVclAddrEntry 1 }

   atmVclAddrType     OBJECT-TYPE
       SYNTAX       INTEGER {
           callingParty(1),
           calledParty(2)
       }
       MAX-ACCESS   read-create
       STATUS       current
       DESCRIPTION
            "The type of ATM Address represented by the object
            atmVclAddrAddr.  Choices are either the calling party ATM
            address or the called party ATM address."
       ::= { atmVclAddrEntry 2 }

   atmVclAddrRowStatus     OBJECT-TYPE
       SYNTAX       RowStatus
       MAX-ACCESS   read-create
       STATUS       current
       DESCRIPTION
           "This object is used to create or destroy an
           entry from this table.  Note that the manager entity
           can only destroy the PVC VCLs."
       ::= { atmVclAddrEntry 3 }

  -- 8. ATM Address to VPI/VCI Mapping Table
  --
  -- This table provides an alternative way to access
  -- a row in the atmVclAddrTable by using
  -- an ATM address as an index, instead of
  -- the ifIndex

    atmAddrVclTable   OBJECT-TYPE
        SYNTAX       SEQUENCE OF AtmAddrVclEntry
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION
            "This table provides an alternative way to retrieve the
            atmVclTable.  This table can be used to retrieve the
            indexing to the atmVclTable by an ATM address."

        ::= { atm2MIBObjects 8 }

    atmAddrVclEntry    OBJECT-TYPE
        SYNTAX       AtmAddrVclEntry
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION
            "Each entry in this table represents an entry in the
            atmVclTable of the ATM-MIB by its ATM address.  The ATM
            address is either the calling or called party ATM address
            of the call.  Entries in this table are read only.
            They show up when entries are created in the
            atmVclAddrTable."
        REFERENCE
            "Tesink, K., Editor, Definitions of Managed Objects
             for ATM Management, RFC 2515, Bell Communications
             Research, February, 1999."
        INDEX { atmVclAddrAddr, atmAddrVclAtmIfIndex,
                atmAddrVclVpi, atmAddrVclVci }
        ::= { atmAddrVclTable 1 }

    AtmAddrVclEntry ::=
        SEQUENCE {
            atmAddrVclAtmIfIndex    InterfaceIndex,
            atmAddrVclVpi           AtmVpIdentifier,
            atmAddrVclVci           AtmVcIdentifier,
            atmAddrVclAddrType      INTEGER
            }
    atmAddrVclAtmIfIndex     OBJECT-TYPE
        SYNTAX       InterfaceIndex
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION
            "The interface index of the ATM interface to which this
             VCL pertains.  This object combined with the
             atmAddrVclVpi and atmAddrVclVci objects serves as an
             index to the atmVclTable."
        ::= { atmAddrVclEntry 1 }

    atmAddrVclVpi     OBJECT-TYPE
        SYNTAX       AtmVpIdentifier
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION
            "The VPI value of the VCL.  This object combined with the
            atmAddrVclAtmIfIndex and atmAddrVclVci objects serves as
            an index to the atmVclTable."
        ::= { atmAddrVclEntry 2 }

    atmAddrVclVci     OBJECT-TYPE
        SYNTAX       AtmVcIdentifier
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION
            "The VCI value of the VCL.  This object combined with the
            atmAddrVclAtmIfIndex and atmAddrVclVpi objects serves as
            an index to the atmVclTable."
        ::= { atmAddrVclEntry 3 }

   atmAddrVclAddrType     OBJECT-TYPE
       SYNTAX       INTEGER {
           callingParty(1),
           calledParty(2) }
       MAX-ACCESS   read-only
       STATUS       current
       DESCRIPTION
            "The type of ATM Address represented by the object
            atmVclAddrAddr.  Choices are either calling party address
            or called party address."
       ::= { atmAddrVclEntry 4 }

  -- 9. ATM VPL Statistics Table

    atmVplStatTable   OBJECT-TYPE
        SYNTAX       SEQUENCE OF AtmVplStatEntry
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION
            "This table contains all statistics counters per VPL.  It is
            used to monitor the usage of the VPL in terms of incoming
            cells and outgoing cells."
        ::= { atm2MIBObjects 9 }

    atmVplStatEntry    OBJECT-TYPE
        SYNTAX       AtmVplStatEntry
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION
            "Each entry in this table represents a VPL."
        INDEX { ifIndex, atmVplVpi }
        ::= { atmVplStatTable 1 }

    AtmVplStatEntry ::=
        SEQUENCE {
            atmVplStatTotalCellIns      Counter32,
            atmVplStatClp0CellIns       Counter32,
            atmVplStatTotalDiscards     Counter32,
            atmVplStatClp0Discards      Counter32,
            atmVplStatTotalCellOuts     Counter32,
            atmVplStatClp0CellOuts      Counter32,
            atmVplStatClp0Tagged        Counter32
    }

    atmVplStatTotalCellIns OBJECT-TYPE
        SYNTAX       Counter32
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION
            "The total number of valid ATM cells received by this VPL
            including both CLP=0 and CLP=1 cells.  The cells are
            counted prior to the application of the traffic policing."
        ::= { atmVplStatEntry 1 }

    atmVplStatClp0CellIns OBJECT-TYPE
        SYNTAX       Counter32
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION
            "The number of valid ATM cells received by this VPL with
            CLP=0.  The cells are counted prior to the application of
            the traffic policing."
        ::= { atmVplStatEntry 2 }

    atmVplStatTotalDiscards OBJECT-TYPE
```

```
        SYNTAX       Counter32
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION
            "The total number of valid ATM cells discarded by the
            traffic policing entity.  This includes cells originally
            received with CLP=0 and CLP=1."
        ::= { atmVplStatEntry 3 }

    atmVplStatClp0Discards OBJECT-TYPE
        SYNTAX       Counter32
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION
            "The total number of valid ATM cells received with CLP=0 and
            discarded by the traffic policing entity."
        ::= { atmVplStatEntry 4 }

    atmVplStatTotalCellOuts OBJECT-TYPE
        SYNTAX       Counter32
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION
            "The total number of valid ATM cells transmitted  by this
            VPL.  This includes both CLP=0 and CLP=1 cells."
        ::= { atmVplStatEntry 5 }

    atmVplStatClp0CellOuts OBJECT-TYPE
        SYNTAX       Counter32
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION
            "The total number of valid ATM cells transmitted with CLP=0
            by this VPL."
        ::= { atmVplStatEntry 6 }

    atmVplStatClp0Tagged OBJECT-TYPE
        SYNTAX       Counter32
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION
            "The total number of valid ATM cells tagged by the traffic
            policing entity from CLP=0 to CLP=1."
        ::= { atmVplStatEntry 7 }

  -- 10. ATM Logical Port Configuration Table

    atmVplLogicalPortTable OBJECT-TYPE
```

```
        SYNTAX       SEQUENCE OF AtmVplLogicalPortEntry
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION
            "Indicates whether the VPL is an ATM Logical Port interface
            (ifType=80)."
        ::= { atm2MIBObjects 10 }

    atmVplLogicalPortEntry OBJECT-TYPE
        SYNTAX       AtmVplLogicalPortEntry
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION
            "An entry with information about the ATM Logical Port
            interface."
        AUGMENTS { atmVplEntry }
        ::= { atmVplLogicalPortTable 1 }

    AtmVplLogicalPortEntry ::=
        SEQUENCE {
            atmVplLogicalPortDef        INTEGER,
            atmVplLogicalPortIndex      InterfaceIndexOrZero
            }

    atmVplLogicalPortDef   OBJECT-TYPE
        SYNTAX       INTEGER {
                        notLogicalIf(1),
                        isLogicalIf(2)
                        }
        MAX-ACCESS   read-create
        STATUS       current
        DESCRIPTION
            "Indicates whether the VPC at this VPL interface is an ATM
            Logical Port interface."
        DEFVAL { notLogicalIf }
        ::= { atmVplLogicalPortEntry 1 }

    atmVplLogicalPortIndex OBJECT-TYPE
        SYNTAX       InterfaceIndexOrZero
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION
            "The ifTable index of the ATM logical port interface
            associated with this VPL. The distinguished value of zero
            indicates that the agent has not (yet) assigned such an
            ifTable Index. The zero value must be assigned by the agent
            if the value of atmVplLogicalPortDef is set to notLogicalIf,
            or if the VPL row is not active."
        ::= { atmVplLogicalPortEntry 2 }

  -- 11. ATM VCL Statistics Table

    atmVclStatTable   OBJECT-TYPE
        SYNTAX       SEQUENCE OF AtmVclStatEntry
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION
            "This table contains all statistics counters per VCL.  It is
            used to monitor the usage of the VCL in terms of incoming
            cells and outgoing cells."
        ::= { atm2MIBObjects 11 }

    atmVclStatEntry    OBJECT-TYPE
        SYNTAX       AtmVclStatEntry
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION
            "Each entry in this table represents a VCL."
        INDEX { ifIndex, atmVclVpi, atmVclVci }
        ::= { atmVclStatTable 1 }

    AtmVclStatEntry ::=
        SEQUENCE {
            atmVclStatTotalCellIns      Counter32,
            atmVclStatClp0CellIns       Counter32,
            atmVclStatTotalDiscards     Counter32,
            atmVclStatClp0Discards      Counter32,
            atmVclStatTotalCellOuts     Counter32,
            atmVclStatClp0CellOuts      Counter32,
            atmVclStatClp0Tagged        Counter32
        }

    atmVclStatTotalCellIns OBJECT-TYPE
        SYNTAX       Counter32
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION
            "The total number of valid ATM cells received by this VCL
            including both CLP=0 and CLP=1 cells.  The cells are counted
            prior to the application of the traffic policing."
        ::= { atmVclStatEntry 1 }

    atmVclStatClp0CellIns OBJECT-TYPE
        SYNTAX       Counter32
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION
            "The number of valid ATM cells received by this VCL with
            CLP=0.  The cells are counted prior to the application of
            the traffic policing."
        ::= { atmVclStatEntry 2 }

    atmVclStatTotalDiscards OBJECT-TYPE
        SYNTAX       Counter32
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION
            "The total number of valid ATM cells discarded by the
            traffic policing entity.  This includes cells originally
            received with CLP=0 and CLP=1."
        ::= { atmVclStatEntry 3 }

    atmVclStatClp0Discards OBJECT-TYPE
        SYNTAX       Counter32
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION
            "The total number of valid ATM cells received with CLP=0
             and discarded by the traffic policing entity."
        ::= { atmVclStatEntry 4 }

    atmVclStatTotalCellOuts OBJECT-TYPE
        SYNTAX       Counter32
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION
            "The total number of valid ATM cells transmitted  by this
            VCL.  This includes both CLP=0 and CLP=1 cells."
        ::= { atmVclStatEntry 5 }

    atmVclStatClp0CellOuts OBJECT-TYPE
        SYNTAX       Counter32
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION
            "The total number of valid ATM cells transmitted with CLP=0
            by this VCL."
        ::= { atmVclStatEntry 6 }

    atmVclStatClp0Tagged OBJECT-TYPE
        SYNTAX       Counter32
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION
```

```
            "The total number of valid ATM cells tagged by the traffic
            policing entity from CLP=0 to CLP=1."
        ::= { atmVclStatEntry 7 }

  -- 12. ATM AAL5 per-VCC Statistics Table

    atmAal5VclStatTable   OBJECT-TYPE
        SYNTAX       SEQUENCE OF AtmAal5VclStatEntry
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION
            "This table provides a collection of objects providing AAL5
            configuration and performance statistics of a VCL."
        ::= { atm2MIBObjects 12 }

    atmAal5VclStatEntry    OBJECT-TYPE
        SYNTAX       AtmAal5VclStatEntry
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION
            "Each entry in this table represents an AAL5 VCL, and
             is indexed by ifIndex values of AAL5 interfaces and
             the associated VPI/VCI values."
        INDEX { ifIndex, atmVclVpi, atmVclVci }
        ::= { atmAal5VclStatTable 1 }

    AtmAal5VclStatEntry ::=
        SEQUENCE {
           atmAal5VclInPkts        Counter32,
           atmAal5VclOutPkts       Counter32,
           atmAal5VclInOctets      Counter32,
           atmAal5VclOutOctets     Counter32
        }

    atmAal5VclInPkts      OBJECT-TYPE
        SYNTAX       Counter32
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION
            "The number of AAL5 CPCS PDUs received on the AAL5 VCC at
            the interface identified by the ifIndex."
        ::= { atmAal5VclStatEntry 1 }

    atmAal5VclOutPkts   OBJECT-TYPE
        SYNTAX       Counter32
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION
```

```
            "The number of AAL5 CPCS PDUs transmitted on the AAL5 VCC
             at the interface identified by the ifIndex."
        ::= { atmAal5VclStatEntry 2 }

    atmAal5VclInOctets   OBJECT-TYPE
        SYNTAX       Counter32
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION
            "The number of octets contained in AAL5 CPCS PDUs received
            on the AAL5 VCC at the interface identified by the ifIndex."
        ::= { atmAal5VclStatEntry 3 }

    atmAal5VclOutOctets   OBJECT-TYPE
        SYNTAX       Counter32
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION
            "The number of octets contained in AAL5 CPCS PDUs
            transmitted on the AAL5 VCC at the interface identified by
            the ifIndex."
        ::= { atmAal5VclStatEntry 4 }

  -- 13. ATM VC General Information Table

    atmVclGenTable OBJECT-TYPE
        SYNTAX       SEQUENCE OF AtmVclGenEntry
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION
            "General Information for each VC."
        ::= { atm2MIBObjects 13 }

    atmVclGenEntry OBJECT-TYPE
        SYNTAX       AtmVclGenEntry
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION
            "An entry with general information about the ATM VC."

        AUGMENTS { atmVclEntry }
        ::= { atmVclGenTable 1 }

    AtmVclGenEntry ::=
        SEQUENCE {
          atmVclGenSigDescrIndex      AtmSigDescrParamIndex
        }
    atmVclGenSigDescrIndex   OBJECT-TYPE
        SYNTAX       AtmSigDescrParamIndex
        MAX-ACCESS   read-create
        STATUS       current
        DESCRIPTION
            "The value of this object identifies the row in the ATM
            Signalling Descriptor Parameter Table which applies to this
            VCL."
        ::= { atmVclGenEntry 1 }

  -- 14. ATM Interface Configuration Extension Table

atmInterfaceExtTable    OBJECT-TYPE
    SYNTAX       SEQUENCE OF AtmInterfaceExtEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "This table contains ATM interface configuration and monitoring
       information not defined in the atmInterfaceConfTable from the
       ATM-MIB.  This includes the type of connection setup procedures,
       ILMI information, and information on the VPI/VCI range."
    REFERENCE
        "Tesink, K., Editor, Definitions of Managed Objects
         for ATM Management, RFC 2515, Bell Communications
         Research, February, 1999."
    ::= { atm2MIBObjects 14 }

atmInterfaceExtEntry    OBJECT-TYPE
    SYNTAX       AtmInterfaceExtEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "An entry extends the atmInterfaceConfEntry defined in the ATM-
       MIB.  Each entry corresponds to an ATM interface."
    REFERENCE
        "Tesink, K., Editor, Definitions of Managed Objects
         for ATM Management, RFC 2515, Bell Communications
         Research, February, 1999."
    AUGMENTS { atmInterfaceConfEntry }
    ::= { atmInterfaceExtTable 1 }

AtmInterfaceExtEntry    ::= SEQUENCE  {
        atmIntfConfigType                  AtmInterfaceType,
        atmIntfActualType                  AtmInterfaceType,
        atmIntfConfigSide                  INTEGER,
        atmIntfActualSide                  INTEGER,
        atmIntfIlmiAdminStatus             BITS,
        atmIntfIlmiOperStatus              BITS,
        atmIntfIlmiFsmState                    INTEGER,
        atmIntfIlmiEstablishConPollIntvl    Integer32,
        atmIntfIlmiCheckConPollIntvl        Integer32,
        atmIntfIlmiConPollInactFactor       Integer32,
        atmIntfIlmiPublicPrivateIndctr      INTEGER,
        atmInterfaceConfMaxSvpcVpi          INTEGER,
        atmInterfaceCurrentMaxSvpcVpi       INTEGER,
        atmInterfaceConfMaxSvccVpi          INTEGER,
        atmInterfaceCurrentMaxSvccVpi       INTEGER,
        atmInterfaceConfMinSvccVci          INTEGER,
        atmInterfaceCurrentMinSvccVci       INTEGER,
        atmIntfSigVccRxTrafficDescrIndex
                                   AtmTrafficDescrParamIndex,
        atmIntfSigVccTxTrafficDescrIndex
                                   AtmTrafficDescrParamIndex,
        atmIntfPvcFailures                  Counter32,
        atmIntfCurrentlyFailingPVpls        Gauge32,
        atmIntfCurrentlyFailingPVcls        Gauge32,
        atmIntfPvcFailuresTrapEnable        TruthValue,
        atmIntfPvcNotificationInterval      INTEGER,
        atmIntfLeafSetupFailures            Counter32,
        atmIntfLeafSetupRequests            Counter32 }

atmIntfConfigType     OBJECT-TYPE
    SYNTAX       AtmInterfaceType
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
       "The type of connection setup procedures configured for the ATM
       interface.  Setting this variable to a value of ’other’ is not
       allowed."
    DEFVAL { autoConfig }
    ::= { atmInterfaceExtEntry 1 }

atmIntfActualType     OBJECT-TYPE
    SYNTAX       AtmInterfaceType
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "The type of connection setup procedures currently being used on
       the interface.  This may reflect a manually configured value for
       the interface type, or may be determined by other means such as
       auto-configuration.  A value of ‘autoConfig’ indicates that
       auto-configuration was requested but has not yet been completed."
    ::= { atmInterfaceExtEntry 2 }

atmIntfConfigSide     OBJECT-TYPE
    SYNTAX       INTEGER  {
```

```
                     other(1),
                     user(2),
                     network(3) }
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
       "The configured role of the managed entity as one side of the ATM
       interface.  This value does not apply when the object
       atmIntfConfigType is set to ‘autoConfig’, ‘atmfPnni1Dot0’, or
       ‘atmfBici2Dot0’."
    ::= { atmInterfaceExtEntry 3 }

atmIntfActualSide     OBJECT-TYPE
    SYNTAX       INTEGER  {
                     other(1),
                     user(2),
                     network(3),
                     symmetric(4) }
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "The current role used by the managed entity to represent one
       side of the ATM interface."
    ::= { atmInterfaceExtEntry 4 }

atmIntfIlmiAdminStatus    OBJECT-TYPE
    SYNTAX       BITS { ilmi(0),
                        ilmiAddressRegistration(1),
                        ilmiConnectivity(2),
                        ilmiPvcPvpMgmt(3),
                        ilmiSigVccParamNegotiation(4) }
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
       "Indicates which components of ILMI are administratively enabled
       on this interface.  If the ’ilmi’ bit is not set, then no ILMI
       components are operational.  ILMI components other than auto-
       configuration that are not represented in the value have their
       administrative status determined according to the ’ilmi’ bit.
       The ILMI auto-configuration component is enabled/disabled by the
       atmIntfConfigType object."
    ::= { atmInterfaceExtEntry 5 }

atmIntfIlmiOperStatus    OBJECT-TYPE
    SYNTAX       BITS { ilmi(0),
                        ilmiAddressRegistration(1),
                        ilmiConnectivity(2),
                        ilmiPvcPvpMgmt(3),
                        ilmiSigVccParamNegotiation(4) }
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "Indicates which components of ILMI are operational on this
       interface."
    ::= { atmInterfaceExtEntry 6 }

atmIntfIlmiFsmState    OBJECT-TYPE
    SYNTAX       INTEGER { stopped(1),
                           linkFailing(2),
                           establishing(3),
                           configuring(4),
                           retrievingNetworkPrefixes(5),
                           registeringNetworkPrefixes(6),
                           retrievingAddresses(7),
                           registeringAddresses(8),
                           verifying(9) }
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "Indicates the state of the ILMI Finite State Machine associated
       with this interface."

        REFERENCE
            "ATM Forum, Integrated Local Management Interface
             (ILMI) Specification, Version 4.0, af-ilmi-0065.000,
             September 1996, Appendix 1"
    ::= { atmInterfaceExtEntry 7 }

atmIntfIlmiEstablishConPollIntvl    OBJECT-TYPE
    SYNTAX       Integer32 (1..65535)
    UNITS        "seconds"
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
       "The amount of time S between successive transmissions of ILMI
       messages on this interface for the purpose of detecting
       establishment of ILMI connectivity."
    REFERENCE
       "ATM Forum, Integrated Local Management Interface
        (ILMI) Specification, Version 4.0, af-ilmi-0065.000,
        September 1996, Section 8.3.1"
    DEFVAL { 1 }
    ::= { atmInterfaceExtEntry 8 }

atmIntfIlmiCheckConPollIntvl    OBJECT-TYPE
    SYNTAX       Integer32 (0..65535)
    UNITS        "seconds"
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
       "The amount of time T between successive transmissions of ILMI
       messages on this interface for the purpose of detecting loss of
       ILMI connectivity.  The distinguished value zero disables ILMI
       connectivity procedures on this interface."
    REFERENCE
       "ATM Forum, Integrated Local Management Interface
        (ILMI) Specification, Version 4.0, af-ilmi-0065.000,
        September 1996, Section 8.3.1"
    DEFVAL { 5 }
    ::= { atmInterfaceExtEntry 9 }

atmIntfIlmiConPollInactFactor    OBJECT-TYPE
    SYNTAX       Integer32 (0..65535)
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
       "The number K of consecutive polls on this interface for which no
       ILMI response message is received before ILMI connectivity is
       declared lost."
    REFERENCE
       "ATM Forum, Integrated Local Management Interface
        (ILMI) Specification, Version 4.0, af-ilmi-0065.000,
        September 1996, Section 8.3.1"
    DEFVAL { 4 }
    ::= { atmInterfaceExtEntry 10 }

atmIntfIlmiPublicPrivateIndctr    OBJECT-TYPE
    SYNTAX       INTEGER {
                           other(1),
                           public(2),
                           private(3)
                           }
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
       "Specifies whether this end of the interface is advertised in
       ILMI as a device of the ‘public’ or ‘private’ type."
    DEFVAL { private }
    ::= { atmInterfaceExtEntry 11 }

atmInterfaceConfMaxSvpcVpi    OBJECT-TYPE
    SYNTAX       INTEGER  (0..4095)
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
       "The maximum VPI that the signalling stack on the ATM interface
       is configured to support for allocation to switched virtual path
       connections."
    ::= { atmInterfaceExtEntry 12 }

atmInterfaceCurrentMaxSvpcVpi    OBJECT-TYPE
    SYNTAX       INTEGER  (0..4095)
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "The maximum VPI that the signalling stack on the ATM interface
       may currently allocate to switched virtual path connections.
       This value is the minimum of atmInterfaceConfMaxSvpcVpi, and the
       atmInterfaceMaxSvpcVpi of the interface's UNI/NNI peer.

       If the interface does not negotiate with its peer to determine
       the maximum VPI that can be allocated to SVPCs on the interface,
       then the value of this object must equal
       atmInterfaceConfMaxSvpcVpi. "
    ::= { atmInterfaceExtEntry 13 }

atmInterfaceConfMaxSvccVpi    OBJECT-TYPE
    SYNTAX       INTEGER  (0..4095)
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
       "The maximum VPI that the signalling stack on the ATM interface
       is configured to support for allocation to switched virtual
       channel connections."
    ::= { atmInterfaceExtEntry 14 }

atmInterfaceCurrentMaxSvccVpi    OBJECT-TYPE
    SYNTAX       INTEGER  (0..4095)
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "The maximum VPI that the signalling stack on the ATM interface
       may currently allocate to switched virtual channel connections.
       This value is the minimum of atmInterfaceConfMaxSvccVpi, and the
       atmInterfaceConfMaxSvccVpi of the interface's UNI/NNI peer.

       If the interface does not negotiate with its peer to determine
       the maximum VPI that can be allocated to SVCCs on the interface,
       then the value of this object must equal
       atmInterfaceConfMaxSvccVpi."
    ::= { atmInterfaceExtEntry 15 }
```

```
atmInterfaceConfMinSvccVci    OBJECT-TYPE
    SYNTAX       INTEGER  (0..65535)
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
       "The minimum VCI that the signalling stack on the ATM interface
       is configured to support for allocation to switched virtual
       channel connections."
    ::= { atmInterfaceExtEntry 16 }

atmInterfaceCurrentMinSvccVci    OBJECT-TYPE
    SYNTAX       INTEGER  (0..65535)
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "The minimum VCI that the signalling stack on the ATM interface
       may currently allocate to switched virtual channel connections.
       This value is the maximum of atmInterfaceConfMinSvccVci, and the
       atmInterfaceConfMinSvccVci of the interface’s UNI/NNI peer.
       If the interface does not negotiate with its peer to determine
       the minimum VCI that can be allocated to SVCCs on the interface,
       then the value of this object must equal
       atmInterfaceConfMinSvccVci."
    ::= { atmInterfaceExtEntry 17 }

atmIntfSigVccRxTrafficDescrIndex    OBJECT-TYPE
    SYNTAX       AtmTrafficDescrParamIndex
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
       "This object identifies the row in the atmTrafficDescrParamTable
       used during ILMI auto-configuration to specify the advertised
       signalling VCC traffic parameters for the receive direction.
       The traffic descriptor resulting from ILMI auto-configuration of
       the signalling VCC is indicated in the atmVclTable."
    ::= { atmInterfaceExtEntry 18 }

atmIntfSigVccTxTrafficDescrIndex    OBJECT-TYPE
    SYNTAX       AtmTrafficDescrParamIndex
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
       "This object identifies the row in the atmTrafficDescrParamTable
       used during ILMI auto-configuration to specify the advertised
       signalling VCC traffic parameters for the transmit direction.
       The traffic descriptor resulting from ILMI auto-configuration of
       the signalling VCC is indicated in the atmVclTable."
    ::= { atmInterfaceExtEntry 19 }
```

```
atmIntfPvcFailures    OBJECT-TYPE
    SYNTAX       Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "The number of times the operational status of a PVPL or PVCL on
       this interface has gone down."
    ::= { atmInterfaceExtEntry 20 }

atmIntfCurrentlyFailingPVpls    OBJECT-TYPE
    SYNTAX       Gauge32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "The current number of VPLs on this interface for which there is
       an active row in the atmVplTable having an atmVplConnKind value
       of ‘pvc’ and an atmVplOperStatus with a value other than ‘up’."
    ::= { atmInterfaceExtEntry 21 }

atmIntfCurrentlyFailingPVcls    OBJECT-TYPE
    SYNTAX       Gauge32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "The current number of VCLs on this interface for which there is
       an active row in the atmVclTable having an atmVclConnKind value
       of ‘pvc’ and an atmVclOperStatus with a value other than ‘up’."
    ::= { atmInterfaceExtEntry 22 }

atmIntfPvcFailuresTrapEnable    OBJECT-TYPE
    SYNTAX       TruthValue
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
       "Allows the generation of traps in response to PVCL or PVPL
       failures on this interface."
    DEFVAL { false }
    ::= { atmInterfaceExtEntry 23 }

atmIntfPvcNotificationInterval    OBJECT-TYPE
    SYNTAX       INTEGER (1..3600)
    UNITS        "seconds"
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
       "The minimum interval between the sending of
       atmIntfPvcFailuresTrap notifications for this interface."
    DEFVAL { 30 }
    ::= { atmInterfaceExtEntry 24 }

atmIntfLeafSetupFailures    OBJECT-TYPE
    SYNTAX       Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "Number of setup failures.  For root, this is the number of
       rejected setup requests and for leaf, this is the number of setup
       failure received."
    ::= { atmInterfaceExtEntry 25 }

atmIntfLeafSetupRequests    OBJECT-TYPE
    SYNTAX       Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
       "Number of setup requests.  For root, this includes both incoming
       setup request and root intiated setup requests."
    ::= { atmInterfaceExtEntry 26 }

  -- 15. ATM ILMI Service Registry Table

atmIlmiSrvcRegTable OBJECT-TYPE
    SYNTAX       SEQUENCE OF AtmIlmiSrvcRegEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "This table contains a list of all the ATM network services known
       by this device.

       The characteristics of these services are made available through
       the ILMI, using the ILMI general-purpose service registry MIB.
       These services may be made available to all ATM interfaces
       (atmIlmiSrvcRegIndex = 0) or to some specific ATM interfaces only
       (atmIlmiSrvcRegIndex = ATM interface index)."
    ::= { atm2MIBObjects 15 }

atmIlmiSrvcRegEntry OBJECT-TYPE
    SYNTAX       AtmIlmiSrvcRegEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "Information about a single service provider that is available to
       the user-side of an adjacent device through the ILMI.

       Implementors need to be aware that if the size of the
       atmIlmiSrvcRegServiceID exceeds 112 sub-identifiers then OIDs of
       column instances in this table will have more than 128 sub-
       identifiers and cannot be accessed using SNMPv1, SNMPv2, or
       SNMPv3."

    INDEX { atmIlmiSrvcRegIndex,
            atmIlmiSrvcRegServiceID,
            atmIlmiSrvcRegAddressIndex }
    ::= { atmIlmiSrvcRegTable 1 }

AtmIlmiSrvcRegEntry ::= SEQUENCE {
           atmIlmiSrvcRegIndex             InterfaceIndexOrZero,
           atmIlmiSrvcRegServiceID         OBJECT IDENTIFIER,
           atmIlmiSrvcRegAddressIndex      INTEGER,
           atmIlmiSrvcRegATMAddress        AtmAddr,
           atmIlmiSrvcRegParm1             OCTET STRING,
           atmIlmiSrvcRegRowStatus         RowStatus
           }

atmIlmiSrvcRegIndex OBJECT-TYPE
    SYNTAX       InterfaceIndexOrZero
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "The ATM interface where the service defined in this entry can be
       made available to an ATM device attached to this interface.

       The value of 0 has a special meaning: when an ATM service is
       defined in an entry whose atmIlmiSrvcRegIndex is zero, the ATM
       service is available to ATM devices connected to any ATM
       interface.  (default value(s)).

       When the user-side of an adjacent device queries the content of
       the ILMI service registry MIB (using the ILMI protocol), the
       local network-side responds with the ATM services defined in
       atmIlmiSrvcRegTable entries, provided that these entries are
       indexed by:

       - the corresponding ifIndex value (atmIlmiSrvcRegIndex
         equal to the ifIndex of the interface to which the
         adjacent device is connected) - zero (atmIlmiSrvcRegIndex=0)."
    ::= { atmIlmiSrvcRegEntry 1 }

atmIlmiSrvcRegServiceID OBJECT-TYPE
    SYNTAX       OBJECT IDENTIFIER
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "This is the service identifier which uniquely identifies the
```

```
          type of service at the address provided in the table.  The object
          identifiers for the LAN Emulation Configuration Server (LECS) and
          the ATM Name Server (ANS) are defined in the ATM Forum ILMI
          Service Registry MIB.  The object identifiers for the ATMARP
          Server, the Multicast Address Resolution Server (MARS), and the
          NHRP Server (NHS) are defined in RFC 2601, RFC 2602, and RFC
          2603, respectively."
       ::= { atmIlmiSrvcRegEntry 2 }

   atmIlmiSrvcRegAddressIndex OBJECT-TYPE
       SYNTAX       INTEGER (1..2147483647)
       MAX-ACCESS   not-accessible
       STATUS       current
       DESCRIPTION
          "An arbitrary integer to differentiate multiple rows containing
          different ATM addresses for the same service on the same
          interface.  This number need NOT be the same as the corresponding
          ILMI atmfSrvcRegAddressIndex MIB object."
       ::= { atmIlmiSrvcRegEntry 3 }

   atmIlmiSrvcRegATMAddress OBJECT-TYPE
       SYNTAX       AtmAddr
       MAX-ACCESS   read-create
       STATUS       current
       DESCRIPTION
          "This is the full address of the service.  The user-side of the
          adjacent device may use this address to establish a connection
          with the service."
       ::= { atmIlmiSrvcRegEntry 4 }

   atmIlmiSrvcRegParm1 OBJECT-TYPE
       SYNTAX       OCTET STRING (SIZE(1..255))
       MAX-ACCESS   read-create
       STATUS       current
       DESCRIPTION
          "An octet string used according to the value of
          atmIlmiSrvcRegServiceID."
       ::= { atmIlmiSrvcRegEntry 5 }

   atmIlmiSrvcRegRowStatus OBJECT-TYPE
       SYNTAX       RowStatus
       MAX-ACCESS   read-create
       STATUS       current
       DESCRIPTION
          "This object is used to create or destroy an entry from this
          table."
       ::= { atmIlmiSrvcRegEntry 6 }
```

```
  -- 16. ILMI Network Prefix Table

atmIlmiNetworkPrefixTable    OBJECT-TYPE
    SYNTAX       SEQUENCE OF AtmIlmiNetworkPrefixEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "A table specifying per-interface network prefix(es) supplied by
       the network side of the UNI during ILMI address registration.
       When no network prefixes are specified for a particular
       interface, one or more network prefixes based on the switch
       address(es) may be used for ILMI address registration."
    ::= { atm2MIBObjects 16 }

atmIlmiNetworkPrefixEntry OBJECT-TYPE
    SYNTAX       AtmIlmiNetworkPrefixEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "Information about a single network prefix supplied by the
       network side of the UNI during ILMI address registration.  Note
       that the index variable atmIlmiNetPrefixPrefix is a variable-
       length string, and as such the rule for variable-length strings
       in section 7.7 of RFC 2578 applies."
    INDEX   { ifIndex,
              atmIlmiNetPrefixPrefix }
    ::= { atmIlmiNetworkPrefixTable 1 }

AtmIlmiNetworkPrefixEntry ::=
    SEQUENCE {
           atmIlmiNetPrefixPrefix       AtmIlmiNetworkPrefix,
           atmIlmiNetPrefixRowStatus    RowStatus
    }

atmIlmiNetPrefixPrefix    OBJECT-TYPE
    SYNTAX       AtmIlmiNetworkPrefix
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
       "The network prefix specified for use in ILMI address
       registration."
    ::= { atmIlmiNetworkPrefixEntry 1 }

atmIlmiNetPrefixRowStatus    OBJECT-TYPE
    SYNTAX       RowStatus
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
```

```
           "Used to create, delete, activate and de-activate network
           prefixes used in ILMI address registration."

        ::= { atmIlmiNetworkPrefixEntry 2 }

      -- 17. ATM Switch Address Table

    atmSwitchAddressTable    OBJECT-TYPE
        SYNTAX       SEQUENCE OF AtmSwitchAddressEntry
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION
           "This table contains one or more ATM endsystem addresses on a
           per-switch basis.  These addresses are used to identify the
           switch.  When no ILMI network prefixes are configured for certain
           interfaces, network prefixes based on the switch address(es) may
           be used for ILMI address registration."
        ::= { atm2MIBObjects 17 }

    atmSwitchAddressEntry    OBJECT-TYPE
        SYNTAX       AtmSwitchAddressEntry
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION
           "An entry in the ATM Switch Address table."
        INDEX { atmSwitchAddressIndex }
        ::= { atmSwitchAddressTable 1 }

    AtmSwitchAddressEntry ::=
        SEQUENCE {
                   atmSwitchAddressIndex           Integer32,
                   atmSwitchAddressAddress         OCTET STRING,
                   atmSwitchAddressRowStatus       RowStatus
        }

    atmSwitchAddressIndex   OBJECT-TYPE
        SYNTAX       Integer32 (1..65535)
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION
           "An arbitrary index used to enumerate the ATM endsystem addresses
           for this switch."
        ::= { atmSwitchAddressEntry 1 }

    atmSwitchAddressAddress   OBJECT-TYPE
        SYNTAX       OCTET STRING (SIZE(13|20))
        MAX-ACCESS   read-create
        STATUS       current
        DESCRIPTION
           "An ATM endsystem address or address prefix used to identify this
           switch.  When no ESI or SEL field is specified, the switch may
           generate the ESI and SEL fields automatically to obtain a
           complete 20-byte ATM endsystem address."
        ::= { atmSwitchAddressEntry 2 }

    atmSwitchAddressRowStatus OBJECT-TYPE
        SYNTAX       RowStatus
        MAX-ACCESS   read-create
        STATUS       current
        DESCRIPTION
           "Used to create, delete, activate, and de-activate addresses used
           to identify this switch."
        ::= { atmSwitchAddressEntry 3 }

      -- 18. ATM VP Cross-Connect Extension Table

    atmVpCrossConnectXTable OBJECT-TYPE
        SYNTAX       SEQUENCE OF AtmVpCrossConnectXEntry
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION
           "This table contains one row per VP Cross-Connect represented in
           the atmVpCrossConnectTable."
        ::= { atm2MIBObjects 18 }

    atmVpCrossConnectXEntry       OBJECT-TYPE
        SYNTAX       AtmVpCrossConnectXEntry
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION
           "Information about a particular ATM VP Cross-Connect.
           Each entry provides an two objects that name the Cross-Connect.
           One is assigned by the Service User and the other by the Service
           Provider."
        AUGMENTS     { atmVpCrossConnectEntry }
        ::= { atmVpCrossConnectXTable 1 }

    AtmVpCrossConnectXEntry ::= SEQUENCE {
          atmVpCrossConnectUserName       SnmpAdminString,
          atmVpCrossConnectProviderName   SnmpAdminString
    }

    atmVpCrossConnectUserName OBJECT-TYPE
        SYNTAX       SnmpAdminString (SIZE(0..255))
        MAX-ACCESS   read-create
        STATUS       current
        DESCRIPTION
           "This is a service user assigned textual representation of a VPC
           PVC."
        ::= { atmVpCrossConnectXEntry 1 }

    atmVpCrossConnectProviderName OBJECT-TYPE
        SYNTAX       SnmpAdminString (SIZE(0..255))
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION
           "This is a system supplied textual representation of VPC PVC.  It
           is assigned by the service provider."
        ::= { atmVpCrossConnectXEntry 2 }

      -- 19. ATM VC Cross-Connect Extension Table

    atmVcCrossConnectXTable OBJECT-TYPE
        SYNTAX       SEQUENCE OF AtmVcCrossConnectXEntry
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION
           "This table contains one row per VC Cross-Connect represented in
           the atmVcCrossConnectTable."
        ::= { atm2MIBObjects 19 }

    atmVcCrossConnectXEntry       OBJECT-TYPE
        SYNTAX       AtmVcCrossConnectXEntry
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION
           "Information about a particular ATM VC Cross-Connect.
           Each entry provides an two objects that name the Cross-Connect.
           One is assigned by the Service User and the other by the Service
           Provider."
        AUGMENTS   { atmVcCrossConnectEntry }
        ::= { atmVcCrossConnectXTable 1 }

    AtmVcCrossConnectXEntry ::= SEQUENCE {
          atmVcCrossConnectUserName       SnmpAdminString,
          atmVcCrossConnectProviderName   SnmpAdminString
    }

    atmVcCrossConnectUserName OBJECT-TYPE
        SYNTAX       SnmpAdminString (SIZE(0..255))
        MAX-ACCESS   read-create
        STATUS       current
        DESCRIPTION
           "This is a service user assigned textual representation of a VCC
           PVC."
        ::= { atmVcCrossConnectXEntry 1 }

    atmVcCrossConnectProviderName OBJECT-TYPE
        SYNTAX       SnmpAdminString (SIZE(0..255))
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION
           "This is a system supplied textual representation of VCC PVC.  It
           is assigned by the service provider."
        ::= { atmVcCrossConnectXEntry 2 }


      -- 20. Currently Failing PVPL Table

    atmCurrentlyFailingPVplTable    OBJECT-TYPE
        SYNTAX       SEQUENCE OF AtmCurrentlyFailingPVplEntry
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION
           "A table indicating all VPLs for which there is an active row in
           the atmVplTable having an atmVplConnKind value of ‘pvc’ and an
           atmVplOperStatus with a value other than ‘up’."
        ::= { atm2MIBObjects 20 }

    atmCurrentlyFailingPVplEntry    OBJECT-TYPE
        SYNTAX       AtmCurrentlyFailingPVplEntry
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION
           "Each entry in this table represents a VPL for which the
           atmVplRowStatus is ‘active’, the atmVplConnKind is ‘pvc’, and the
           atmVplOperStatus is other than ‘up’."
        INDEX   { ifIndex, atmVplVpi }
        ::= { atmCurrentlyFailingPVplTable 1 }

    AtmCurrentlyFailingPVplEntry ::=
        SEQUENCE {
                   atmCurrentlyFailingPVplTimeStamp    TimeStamp
        }

    atmCurrentlyFailingPVplTimeStamp    OBJECT-TYPE
        SYNTAX       TimeStamp
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION
           "The time at which this PVPL began to fail."
        ::= { atmCurrentlyFailingPVplEntry 1 }
      -- 21. Currently Failing PVCL Table

    atmCurrentlyFailingPVclTable    OBJECT-TYPE
        SYNTAX       SEQUENCE OF AtmCurrentlyFailingPVclEntry
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION
           "A table indicating all VCLs for which there is an active row in
           the atmVclTable having an atmVclConnKind value of ‘pvc’ and an
           atmVclOperStatus with a value other than ‘up’."
        ::= { atm2MIBObjects 21 }

    atmCurrentlyFailingPVclEntry    OBJECT-TYPE
        SYNTAX       AtmCurrentlyFailingPVclEntry
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION
           "Each entry in this table represents a VCL for which the
           atmVclRowStatus is ‘active’, the atmVclConnKind is ‘pvc’, and the
           atmVclOperStatus is other than ‘up’."
        INDEX        { ifIndex, atmVclVpi, atmVclVci }
        ::= { atmCurrentlyFailingPVclTable 1 }

    AtmCurrentlyFailingPVclEntry ::=
        SEQUENCE {
                   atmCurrentlyFailingPVclTimeStamp    TimeStamp
        }

    atmCurrentlyFailingPVclTimeStamp    OBJECT-TYPE
        SYNTAX       TimeStamp
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION
           "The time at which this PVCL began to fail."
        ::= { atmCurrentlyFailingPVclEntry 1 }

      -- ATM PVC Traps

      atmPvcTraps    OBJECT IDENTIFIER ::= { atm2MIBTraps 1 }

      atmPvcTrapsPrefix    OBJECT IDENTIFIER ::= { atmPvcTraps 0 }

      atmIntfPvcFailuresTrap    NOTIFICATION-TYPE
        OBJECTS      { ifIndex, atmIntfPvcFailures,
                       atmIntfCurrentlyFailingPVpls,
                       atmIntfCurrentlyFailingPVcls }
        STATUS       current
        DESCRIPTION
```

```
           "A notification indicating that one or more PVPLs or PVCLs on
           this interface has failed since the last atmPvcFailuresTrap was
           sent.  If this trap has not been sent for the last
           atmIntfPvcNotificationInterval, then it will be sent on the next
           increment of atmIntfPvcFailures."
        ::= { atmPvcTrapsPrefix 1 }

      -- Conformance Information

    atm2MIBConformance   OBJECT IDENTIFIER ::= {atm2MIB 3}

    atm2MIBGroups        OBJECT IDENTIFIER ::= {atm2MIBConformance 1}

    atm2MIBCompliances   OBJECT IDENTIFIER ::= {atm2MIBConformance 2}

      -- Compliance Statements

    atm2MIBCompliance   MODULE-COMPLIANCE
        STATUS       current
        DESCRIPTION
            "The compliance statement for SNMP entities which represent ATM
             interfaces.  The compliance statements are used to determine
             if a particular group or object applies to hosts,
             networks/switches, or both.  The Common group is defined as
             applicable to all three."

        MODULE  -- this module
        MANDATORY-GROUPS  { atmCommonGroup }

      -- Objects in the ATM Switch/Service/Host Group

        GROUP       atmCommonStatsGroup
        DESCRIPTION
            "This group is mandatory for systems that are supporting
             per-VPC or per-VCC counters."

        OBJECT       atmVplLogicalPortDef
        MIN-ACCESS   read-only
        DESCRIPTION
            "This object is mandatory for systems support ATM Logical
             Port interfaces."

        OBJECT       atmIntfSigVccRxTrafficDescrIndex
        DESCRIPTION
            "This object is mandatory for systems that support negotiation
             of signalling VCC traffic parameters through ILMI."

        OBJECT       atmIntfSigVccTxTrafficDescrIndex
        DESCRIPTION
            "This object is mandatory for systems that support negotiation
             of signalling VCC traffic parameters through ILMI."

        OBJECT       atmCurrentlyFailingPVplTimeStamp
        DESCRIPTION
            "This object is optional."

        OBJECT       atmCurrentlyFailingPVclTimeStamp
        DESCRIPTION
            "This object is optional."

        OBJECT       atmIntfLeafSetupFailures
        DESCRIPTION
            "This object is optional."

        OBJECT       atmIntfLeafSetupRequests
        DESCRIPTION
            "This object is optional."

      -- Objects in the ATM Switch/Service Group

        GROUP        atmSwitchServcGroup
        DESCRIPTION
            "This group is mandatory for a Switch/Service that implements
             ATM interfaces."

        OBJECT       atmIfRegAddrRowStatus
        MIN-ACCESS   read-only
        DESCRIPTION
            "Write access is not required, and only one of the six
             enumerated values for the RowStatus textual convention need
             be supported, specifically: active(1)."

        OBJECT       atmSvcVpCrossConnectRowStatus
        MIN-ACCESS   read-only
        DESCRIPTION
            "Write access is not required, and only one of the six
             enumerated values for the RowStatus textual convention need
             be supported, specifically: active(1)"

        OBJECT       atmSvcVcCrossConnectRowStatus
        MIN-ACCESS   read-only
        DESCRIPTION
            "Write access is not required, and only one of the six
             enumerated values for the RowStatus textual convention need
             be supported, specifically: active(1)"
      -- Objects in the ATM Switch/Service Signalling Group

        GROUP       atmSwitchServcSigGroup
        DESCRIPTION
            "This group’s write access is not required."

      -- Objects in the ATM Switch/Service Notifications Group

        GROUP        atmSwitchServcNotifGroup
        DESCRIPTION
            "This group is optional for systems implementing support for
             an ATM Switch or an ATM Network Service."

      -- Objects in the ATM Switch Group

        GROUP        atmSwitchGroup
        DESCRIPTION
            "This group is optional for a switch that implements ATM
             interfaces."

      -- Objects in the ATM Service Group

        GROUP        atmServcGroup
        DESCRIPTION
            "This group is mandatory for systems implementing support for
             an ATM Network Service."

      -- Objects in the ATM Host Group

        GROUP        atmHostGroup
        DESCRIPTION
            "This group is mandatory for a Host that implements ATM
             interfaces."

        OBJECT       atmVclAddrType
        MIN-ACCESS   read-only
        DESCRIPTION
            "Write access is not required."

        OBJECT       atmVclAddrRowStatus
        MIN-ACCESS   read-only
        DESCRIPTION
            "Write access is not required, and only one of the six
             enumerated values for the RowStatus textual convention need
             be supported, specifically:  active(1)."

      -- ATM Host Sig Descriptor Parameter Group
```

```
    GROUP        atmHostSigDescrGroup
    DESCRIPTION
        "This group is mandatory for a Host that implements ATM
         interfaces.  Write access is not required for this group."

    ::= { atm2MIBCompliances 1 }

  -- **********************************************

  -- Units of Conformance
  -- Mandatory for ATM hosts and switch/service providers

atmCommonGroup     OBJECT-GROUP

OBJECTS {
    atmSigSSCOPConEvents,
    atmSigSSCOPErrdPdus,
    atmSigDetectSetupAttempts,
    atmSigEmitSetupAttempts,
    atmSigDetectUnavailRoutes,
    atmSigEmitUnavailRoutes,
    atmSigDetectUnavailResrcs,
    atmSigEmitUnavailResrcs,
    atmSigDetectCldPtyEvents,
    atmSigEmitCldPtyEvents,
    atmSigDetectMsgErrors,
    atmSigEmitMsgErrors,
    atmSigDetectClgPtyEvents,
    atmSigEmitClgPtyEvents,
    atmSigDetectTimerExpireds,
    atmSigEmitTimerExpireds,
    atmSigDetectRestarts,
    atmSigEmitRestarts,
    atmSigInEstabls,
    atmSigOutEstabls,
    atmVplLogicalPortDef,
    atmVplLogicalPortIndex,
    atmInterfaceConfMaxSvpcVpi,
    atmInterfaceCurrentMaxSvpcVpi,
    atmInterfaceConfMaxSvccVpi,
    atmInterfaceCurrentMaxSvccVpi,
    atmInterfaceConfMinSvccVci,
    atmInterfaceCurrentMinSvccVci,
    atmIntfSigVccRxTrafficDescrIndex,
    atmIntfSigVccTxTrafficDescrIndex,
    atmIntfPvcFailures,
    atmIntfCurrentlyFailingPVpls,
    atmIntfCurrentlyFailingPVcls,
    atmIntfPvcNotificationInterval,
    atmIntfPvcFailuresTrapEnable,
    atmIntfLeafSetupFailures,
    atmIntfLeafSetupRequests,
    atmIntfConfigType,
    atmIntfActualType,
    atmIntfConfigSide,
    atmIntfActualSide,
    atmIntfIlmiAdminStatus,
    atmIntfIlmiOperStatus,
    atmIntfIlmiFsmState,
    atmIntfIlmiEstablishConPollIntvl,
    atmIntfIlmiCheckConPollIntvl,
    atmIntfIlmiConPollInactFactor,
    atmIntfIlmiPublicPrivateIndctr,
    atmCurrentlyFailingPVplTimeStamp,
    atmCurrentlyFailingPVclTimeStamp
}

  STATUS       current
  DESCRIPTION
    "A collection of objects providing information
    for a Switch/Service/Host that implements
    ATM interfaces."
  ::= { atm2MIBGroups 1 }

atmCommonStatsGroup     OBJECT-GROUP

OBJECTS {
    atmVclStatTotalCellIns,
    atmVclStatClp0CellIns,
    atmVclStatTotalDiscards,
    atmVclStatClp0Discards,
    atmVclStatTotalCellOuts,
    atmVclStatClp0CellOuts,
    atmVclStatClp0Tagged,
    atmVplStatTotalCellIns,
    atmVplStatClp0CellIns,
    atmVplStatTotalDiscards,
    atmVplStatClp0Discards,
    atmVplStatTotalCellOuts,
    atmVplStatClp0CellOuts,
    atmVplStatClp0Tagged
}

  STATUS       current
  DESCRIPTION
    "A collection of objects providing information
```

```
         for a Switch/Service/Host that implements
         ATM VCL and VPL Statistics"
       ::= { atm2MIBGroups 2 }

       atmSwitchServcGroup     OBJECT-GROUP

       OBJECTS {
         atmIlmiSrvcRegATMAddress,
         atmIlmiSrvcRegParm1,
         atmIlmiSrvcRegRowStatus,
         atmIlmiNetPrefixRowStatus,
         atmSvcVpCrossConnectCreationTime,
         atmSvcVpCrossConnectRowStatus,
         atmSvcVcCrossConnectCreationTime,
         atmSvcVcCrossConnectRowStatus,
         atmIfRegAddrAddressSource,
         atmIfRegAddrOrgScope,
         atmIfRegAddrRowStatus}
       STATUS       current
       DESCRIPTION
         "A collection of objects providing information
         for a Switch/Service that implements ATM interfaces."
       ::= { atm2MIBGroups 3 }

       atmSwitchServcSigGroup     OBJECT-GROUP

       OBJECTS {
         atmSigSupportClgPtyNumDel,
         atmSigSupportClgPtySubAddr,
         atmSigSupportCldPtySubAddr,
         atmSigSupportHiLyrInfo,
         atmSigSupportLoLyrInfo,
         atmSigSupportBlliRepeatInd,
         atmSigSupportAALInfo,
         atmSigSupportPrefCarrier}
       STATUS       current
       DESCRIPTION
         "A collection of objects providing information
         for a Switch/Service that implements ATM signalling."
       ::= { atm2MIBGroups 4 }

         atmSwitchServcNotifGroup     NOTIFICATION-GROUP

         NOTIFICATIONS { atmIntfPvcFailuresTrap }
         STATUS       current
         DESCRIPTION
             "A collection of notifications providing information
             for a Switch/Service that implements ATM interfaces."
```

```
    ::= { atm2MIBGroups 5 }

    atmSwitchGroup     OBJECT-GROUP
        OBJECTS {
            atmSwitchAddressAddress,
            atmSwitchAddressRowStatus }
        STATUS       current
        DESCRIPTION
          "A collection of objects providing information
          for an ATM switch."
        ::= { atm2MIBGroups 6 }

    atmServcGroup     OBJECT-GROUP

    OBJECTS {
        atmVpCrossConnectUserName,
        atmVpCrossConnectProviderName,
        atmVcCrossConnectUserName,
        atmVcCrossConnectProviderName }
    STATUS       current
    DESCRIPTION
        "A collection of objects providing information
        for an ATM Network Service."
    ::= { atm2MIBGroups 7 }

    atmHostGroup     OBJECT-GROUP

    OBJECTS {
        atmAal5VclInPkts,
        atmAal5VclOutPkts,
        atmAal5VclInOctets,
        atmAal5VclOutOctets,
        atmVclAddrType,
        atmVclAddrRowStatus,
        atmAddrVclAddrType,
        atmVclGenSigDescrIndex}
    STATUS       current
    DESCRIPTION
      "A collection of objects providing information
      for a Host that implements ATM interfaces."
    ::= { atm2MIBGroups 8 }

    atmHostSigDescrGroup     OBJECT-GROUP

    OBJECTS {
           atmSigDescrParamAalType,
           atmSigDescrParamAalSscsType,
           atmSigDescrParamBhliType,
           atmSigDescrParamBhliInfo,
           atmSigDescrParamBbcConnConf,
           atmSigDescrParamBlliLayer2,
           atmSigDescrParamBlliLayer3,
           atmSigDescrParamBlliPktSize,
           atmSigDescrParamBlliSnapId,
           atmSigDescrParamBlliOuiPid,
           atmSigDescrParamRowStatus}
    STATUS       current
    DESCRIPTION
      "A collection of objects providing information
      for a Host that implements ATM interfaces."
    ::= { atm2MIBGroups 9 }

END
```

6. Acknowledgments

This document is a product of the AToMMIB Working Group. Special thanks go to Gary Hanson of ADC Telecommunications for his quality contributions to this specification.

The authors also like to acknowledge John Flick of HP for his thorough and valuable review of this memo.

7. References

7.1. Normative References

[RFC2515] Tesink, K., Ed., "Definitions of Managed Objects for ATM Management", RFC 2515, February 1999.

[ATM Forum 3.0] ATM Forum, "ATM User-Network Interface Specification, Version 3.0 (UNI 3.0)", September 1993.

[ATM Forum UNI 3.1] ATM Forum, "ATM User-Network Interface Specification, Version 3.1 (UNI 3.1)", September 1994.

[ATM Forum LANE] ATM Forum, "LAN Emulation Client Management Specification, Version 1.0", af-lane-0038.000, September 1995.

[RFC1694] Brown, T. and K. Tesink, "Definitions of Managed Objects for SMDS Interfaces using SMIv2", RFC 1694, August 1994.

[ATM Forum ILMI] ATM Forum, "Integrated Local Management Interface (ILMI) Specification, Version 4.0",

[RFC3592] Tesink, K., "Definitions of Managed Objects for the Synchronous Optical Network/Synchronous Digital Hierarchy (SONET/SDH) Interface Type", RFC 3592, September 2003.

[RFC2496] Fowler, D., Ed., "Definitions of Managed Objects for the DS3/E3 Interface Type", RFC 2496, January 1999.

[RFC2578] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M., and S. Waldbusser, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC2579] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M., and S. Waldbusser, "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[RFC2580] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M., and S. Waldbusser, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

[RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.

7.2. Informative References

[RFC3410] Case, J., Mundy, R., Partain, D. and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

8. Security Considerations

There are a number of management objects defined in this MIB module with a MAX-ACCESS clause of read-write and/or read-create. Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations. These are the tables and objects and their sensitivity/vulnerability:

Table	Sensitivity/vulnerability
1. atmSvcVpCrossConnectTable	Deletion of VP cross-connects
2. atmSvcVcCrossConnectTable	Deletion of VC cross-connects
3. atmSigStatTable	Signalling read-only statistics
4. atmSigSupportTable	Signalling configuration params
5. atmSigDescrParamTable	Signalling configuration params
6. atmIfRegisteredAddrTable	Interface address table
7. atmVclAddrTable	VCL/Address mapping table
8. atmAddrVclTable	VCL/Address mapping table (read-only)
9. atmVplStatTable	VPL statistics (read-only)
10. atmVplLogicalPortTable	VPL logical port configuration
11. atmVclStatTable	VCL statistics (read-only)
12. atmAal5VclStatTable	AAL5 statistics (read-only)
13. atmVclGenTable	Signalling configuration
14. atmInterfaceExtTable	Interface configuration
15. atmIlmiSrvcRegTable	ILMI config params
16. atmIlmiNetworkPrefixTable	ILMI config params
17. atmSwitchAddressTable	Switch address info
18. atmVpCrossConnectXTable	VP cross-connect params
19. atmVcCrossConnectXTable	VC cross-connect params
20. atmCurrentlyFailingPVplTable	PVPL status info (read-only)
21. atmCurrentlyFailingPVclTable	PVCL status info (read-only)

Some of the readable objects in this MIB module (i.e., objects with a MAX-ACCESS other than not-accessible) may be considered sensitive or vulnerable in some network environments. It is thus important to control even GET and/or NOTIFY access to these objects and possibly to even encrypt the values of these objects when sending them over the network via SNMP.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPSec), even then, there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

It is RECOMMENDED that implementers consider the security features as provided by the SNMPv3 framework (see [RFC3410], section 8), including full support for the SNMPv3 cryptographic mechanisms (for authentication and privacy).

Further, deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

9. Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF’s procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF Secretariat."

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

10. Authors' Addresses

Faye Ly
Pedestal Networks
6503 Dumbarton Circle
Fremont, CA 94555
USA

Phone (510) 896-2908
EMail: faye@pedestalnetworks.com

Michael Noto
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95134-1706
USA

EMail: mnoto@cisco.com

Andrew Smith
Consultant

EMail: ah_smith@acm.org

Ethan Mickey Spiegel
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95134-1706

Phone: (408) 526-6408
EMail: mspiegel@cisco.com

Kaj Tesink
Telcordia Technologies
331 Newman Springs Road
Red Bank, NJ 07701-7020

Phone: (732) 758-5254
EMail: kaj@research.telcordia.com

11. Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.