

# Understanding ILMI on ATM Interfaces

Document ID: 10449

---

## Introduction

### Before You Begin

Conventions

Prerequisites

Components Used

### Configuring the ILMI PVC

### Understanding the Link MIB

Physical Layer

ATM Layer

Virtual Path Connections (VPCs)

Virtual Channel Connections (VCCs)

### Address Registration on UNI Interfaces

### coldStart Traps

### Errored Responses

### Sample Negotiation

### ILMI Across NNI Interfaces

### Related Information

---

## Introduction

The Interim Local Management Interface (ILMI) is a protocol defined by the ATM Forum for setting and capturing physical layer, ATM layer, virtual path, and virtual circuit parameters on ATM interfaces. ILMI uses simple network management protocol (SNMP) messages without User Datagram Protocol (UDP) and IP, and organizes managed objects into the following four management information bases (MIBs):

- **Textual Conventions MIB** – Defines several textual conventions and object IDs, such as the number of octets for ATM end–system addresses and network prefixes. This document does not cover this MIB.
- **Link Management MIB** – Provides four object groups for all ATM interfaces:
  - ◆ Physical layer – ILMI 4.0 discontinues or "deprecates" earlier physical–layer ILMI values and specifies the use of the standard Interface MIB (RFC 1213). Examples of previous values in this group include:
    - ◇ *atmfTransmissionTypes*, such as *atmfSonetType*, *atmfSonetSTS3c*, *atmfDs3* and *atmfT1*.
    - ◇ *atmfMediaTypes*, such as *atmfMediaUnknownType*, *atmfMediaCoaxCable* and *atmfMediaSingleMode*.
  - ◆ ATM layer – Indicates the number of available bits for virtual path identifier (VPI) and virtual channel identifier (VCI) values in the ATM cell header, maximum number of virtual path connections (VPCs) and virtual channel connections (VCCs) allowed, number of configured permanent virtual paths and permanent virtual channels, and so on.
  - ◆ Virtual path connection – Indicates the up or down status of a VPC and its Quality of Service (QoS) parameters.
  - ◆ Virtual channel connection – Indicates the up or down status of the VCC and its QoS parameters.
- **Address Registration MIB** – Provides an address registration mechanism which allows switches to automatically configure network prefixes in end–systems.

- **Service Registry MIB** – Provides general-purpose service registry for locating ATM network services such as a LAN Emulation Configuration Server (LECS) in LANE.

It is important that you understand ILMI because ATM interfaces use these Simple Network Management Protocol (SNMP) object IDs in network functions such as auto-configuration of a LAN emulation client (LEC) in LANE environments, keepalives, and even permanent virtual circuit (PVC) autodiscovery, which is particularly useful in digital subscriber line (DSL) applications.

This document helps you to understand ILMI and provides some sample debugs to assist you in troubleshooting any problems you encounter.

**Note:** This document focuses on the implementation of ILMI on Cisco routers. For general information on ILMI, please refer to the ILMI Specification on the Approved ATM Forum Specifications page or see the books on the Suggested Reading list of the ATM Technologies page.

## Before You Begin

### Conventions

For more information on document conventions, see the Cisco Technical Tips Conventions.

### Prerequisites

There are no specific prerequisites for this document.

### Components Used

This document is not restricted to specific software and hardware versions.

The information presented in this document was created from devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If you are working in a live network, ensure that you understand the potential impact of any command before using it.

## Configuring the ILMI PVC

When two ATM interfaces run the ILMI protocol, they exchange ILMI packets across the physical connection. These packets consist of SNMP messages as large as 484 octets. ATM interfaces encapsulate these messages in an ATM adaptation layer 5 (AAL5) trailer, segment the packet into cells, and schedule the cells for transmission.

Since ILMI specifies particular values for the AAL5 trailer, we define the encapsulation as ILMI when creating the PVC that will carry the ILMI messages. By default, a PVC with the values of VPI=0 and VCI=16 carries the ILMI messages. We can see in the output of the **show atm ilmi-status** command below that ILMI is using the 0/16 default values.

```
Switch#show atm ilmi-status atm 0/0/0
```

```
Interface : ATM0/0/0 Interface Type : Private UNI (Network-side)
ILMI VCC : (0, 16) ILMI Keepalive : Disabled
ILMI State: UpAndNormal
Peer IP Addr: 10.10.10.4      Peer IF Name: ATM2
Peer MaxVPIbits: 0          Peer MaxVCIBits: 10
Peer MaxVPCs: 0             Peer MaxVCCs: 4096
Peer MaxSvccVpi: 0          Peer MinSvccVci: 0
```

```
Peer MaxSvpcVpi: 0
Configured Prefix(s) :
47.0091.8100.0000.0060.3e5a.8f01
```

On ATM switches such as the Cisco LightStream 1010 and Catalyst 8500 series, an ILMI PVC of 0/16 is configured automatically on each interface. The **show atm vc** command illustrates this automatic configuration. Note how each port's ILMI VC cross-connects to ATM 2/0/0, which is the switch's internal management port. Since ILMI messages are control messages, they must be sent to and processed by the CPU.

```
Switch#show atm vc
Interface VPI VCI Type      X-Interface X-VPI X-VCI Encap Status
ATM0/0/0 0 5      PVC ATM2/0/0 0      39 QSAAL UP
ATM0/0/0 0 16      PVC ATM2/0/0 0      35 ILMI UP
ATM0/0/1 0 5      PVC ATM2/0/0 0      40 QSAAL DOWN
ATM0/0/1 0 16      PVC ATM2/0/0 0      36 ILMI DOWN
ATM0/0/1 4 50     PVC ATM2/0/0 0      230 SNAP DOWN
ATM0/0/2 0 5      PVC ATM2/0/0 0      41 QSAAL UP
ATM0/0/2 0 16      PVC ATM2/0/0 0      37 ILMI UP
ATM0/0/2 0 55     PVC ATM0/0/3 0      50 UP
ATM0/0/2 2 40     PVC ATM2/0/0 0      89 SNAP UP
ATM0/0/2 4 66     PVC ATM2/0/0 0      66 SNAP UP
ATM0/0/3 0 5      PVC ATM2/0/0 0      42 QSAAL UP
ATM0/0/3 0 16      PVC ATM2/0/0 0      38 ILMI UP
```

Optionally, you can configure non-default values for the ILMI PVC using the following procedure. Click here for more information.

```
Switch(config)# interface atm 0/0/0
Switch(config-if)# atm manual-well-known-vc delete
Okay to delete well-known VCs for this interface? [no]: y
Switch(config-if)# atm pvc 1 35 interface atm0 any-vci encap ilmi
Switch(config-if)# end

Switch# show atm vc interface atm 0/0/0
Interface VPI VCI Type      X-Interface X-VPI X-VCI Encap Status
ATM0/0/0 1 35     PVC ATM0      0 150 ILMI UP
```

Caution: It is not recommended to change the default values



**Caution:** It is not recommended to change the default values of the ILMI PVC, as doing so may cause your network to go down. The same PVC should be used between the end device and the switch. Also, manually configuring a different ILMI PVC will make troubleshooting and maintenance more difficult.

## Understanding the Link MIB

The Link MIB of the ILMI MIB consists of the following four groups of objects:

- Physical Layer
- ATM Layer
- Virtual Path Connections (VPCs)
- Virtual Channel Connections (VCCs)

The following sections describe the objects in each group.

### Physical Layer

ILMI 4.0 discontinues or "deprecates" earlier physical-layer ILMI values in the port group and specifies the

use of the standard Interface MIB (RFC 1213). This group also includes objects that allow neighboring systems to maintain a table of adjacent systems to facilitate autodiscovery and tracing of ATM connections.

- atmPortMyIfName
- atmPortMyIfIdentifier
- atmMyIpNmAddress
- atmMySystemIdentifier

The show **atm ilmi-status** command displays the values sent by the peer for these objects.

```
Switch#show atm ilmi-status atm 0/0/0
Interface : ATM0/0/0 Interface Type : Private UNI (Network-side)
ILMI VCC : (0, 16) ILMI Keepalive : Disabled
ILMI State: UpAndNormal
Peer IP Addr: 10.10.10.4 Peer IF Name: ATM2
Peer MaxVPIbits: 0 Peer MaxVCIbits: 10
Peer MaxVPCs: 0 Peer MaxVCCs: 4096
Peer MaxSvccVpi: 0 Peer MinSvccVci: 0
Peer MaxSvpcVpi: 0
Configured Prefix(s) :
47.0091.8100.0000.0060.3e5a.8f01
```

The output of **debug atm ilmi** also captures the values as they are being advertised.

```
1wld: ILMI(ATM0/0/0): KeepAlive disabled
1wld: ILMI: Sending Per-Switch prefix
1wld: ILMI: Registering prefix with end-system 47.0091.8100.0000.0060.3e5a.8f01
1wld: ILMI: The Neighbor's IfName on Intf (ATM0/0/0) is ATM2
1wld: ILMI: The Neighbor's IP on Intf (ATM0/0/0) is 168430084
```

atmMySystemIdentifier is a 48-bit identifier taken from the Institute of Electrical and Electronic Engineers (IEEE) universally-administered MAC address space, which uniquely identifies the ATM device.

## ATM Layer

The following attributes of an ATM interface form the ATM Layer Group, which stores its values in the atmAtmLayerGroup table. Each interface has an atmAtmLayerIndex entry in the table.

- Interface index
- Maximum number of active VPI bits
- Maximum number of active VCI bits
- Maximum number of VPCs
- Maximum number of VCCs
- Number of configured VPCs
- Number of configured VCCs
- Maximum SVPC VPI
- Maximum SVCC VPI
- Minimum SVCC VCI
- ATM interface type
- ATM device Type
- ILMI version
- UNI signaling version
- NNI signaling version

When deciding on the maximum values to use, each side compares the peer's values with its own values. Set the actual number to the highest common value to ensure interoperability.

## Virtual Path Connections (VPCs)

The following attributes of a VPC form the Virtual Path Group, which stores values in the atmVpcGroup table. Each VPC is indexed in the table by an atmVpcPortIndex to identify the physical port and an atmVpcVpi to identify the VPI number.

- Interface index
- VPI value
- Operational status
- Transmit traffic descriptor
- Receive traffic descriptor
- Best effort indicator
- Transmit QoS class
- Receive QoS class
- Service category

## Virtual Channel Connections (VCCs)

The following attributes of a VCC form the virtual channel group, which stores values in the atmVccGroup. Each VCC is indexed in the table by the interface index (atmVccPortIndex), VPI value (atmVccVpi), and VCI value (atmVccVci). Only PVCs are represented in this group, including the well-known or reserved signaling, ilmi and LECS VCCs.

- Interface index
- VPI value
- Operational status
- Transmit traffic descriptor
- Receive traffic descriptor
- Best effort indicator
- Transmit QoS class
- Receive QoS class
- Service category

## Address Registration on UNI Interfaces

The Address Registration MIB provides SNMP objects for the dynamic exchange of ATM address information. This information consists of two tables:

- Network Prefix – Implemented on the ATM end-system via the atmNetPrefixGroup. The ATM switch sends a SetRequest message with the high-order 13-byte prefix configured on that switch port. At initialization, registration of network prefixes occurs first.

```
1wld: ILMI(ATM0/0/0): KeepAlive disabled 1wld: ILMI: Sending  
Per-Switch prefix 1wld: ILMI: Registering prefix with end-system  
47.0091.8100.0000.0060.3e5a.8f01 1wld: ILMI: The Neighbor's IfName  
on Intf (ATM0/0/0) is ATM2 1wld: ILMI: The Neighbor's IP on  
Intf (ATM0/0/0) is 168430084
```

- ATM Address – Implemented on the ATM switch via the atmAddressGroup. The ATM end-system first receives a SetRequest with the network prefix and registers that prefix in its prefix table. Next the ATM end-system combines the prefix with its end-station identifier (ESI) portion and sends a SetRequest with the full 20-byte ATM address. Finally, the ATM switch chooses to register the address in its ATM Address table. The ATM Address table uses two key objects:

- ◆ atmAddressAtmAddress – ATM Address object consists of the full 20–octet private ATM address
- ◆ atmAddressStatus – ATM Address Status object indicates the validity of an ATM address. An ATM end–system configures a new ATM address by sending a SetRequest with the ATM Address Status object set to valid status. An ATM end–system deletes an existing ATM address by sending a SetRequest with the ATM Address Status object set to invalid status.

Both the ATM end–system and the ATM switch need to maintain accurate addressing tables since the addresses are used in Calling Party Number and Called Party Number information element fields of signaling messages sent when switched virtual circuits are being established.

The atmAddressRegistrationAdminStatus object indicates support for the Prefix and Address groups. ILMI 4.0 mandates use of the Prefix and Address groups at a private UNI interface. If the far–end returns a noSuchName error indicating that it is a pre–ILMI 4.0 device, the near–end must assume that the far–end supports address registration. If only one side supports address registration, the ILMI 4.0 specification suggests that the supporting side report a UNI–misconfiguration alarm condition or choose to attempt registration anyway, since the far–end simply should return noSuchName errors to any such registration requests.

ATM Switch (Network–Side)	
Action	When receiving an end–system's SetRequest for an entry in the ATM Address table, the ATM switch validates the advertised address to prevent registration of duplicate addresses.
If validation fails	Responds with a GetResponse containing a badValue error.
If validation succeeds	Responds with a GetResponse indicating noError and updates the address table.

When an ATM end–system de–registers an ATM address, the ATM switch must not clear any connections/calls associated with the de–registered address.

ATM End–System (User–Side)	
Action	Validates a SetRequest for the Network Prefix object.
If validation fails	Responds with a GetResponse containing the appropriate error.
If validation succeeds	Responds with a GetResponse indicating noError and updates the Network Prefix table if the prefix is not already registered.

## coldStart Traps

SNMP uses traps to allow a managed device to report an unusual events back to the management station. It defines several so–called generic traps, one of which is the coldStart trap. ILMI uses the coldStart trap at initialization or re–initialization to clear out or empty any existing entries in the Network Prefix or ATM Address tables. Let's look at how this works:

- ATM end–system sends an ILMI GetNextRequest to read the first instance of the ATM switch's ATM Address Status object. If the response includes a value, the ATM end–system sends a coldStart trap to

tell the ATM switch to initialize the ATM Address table.

- ATM switch sends an ILMI GetNextRequest to read the first instance of the end–system's Network Prefix table. If the response includes a value, the switch sends a coldStart trap to tell the ATM end–system to initialize the Network Prefix table.

In the following sample output, ILMI auto–configuration fails, and ATM interface 1/0/0 sends a coldStart trap to the peer ATM interface.

```
May 11 15:11:19: ILMI: Post trap Config Check Failed. Interface Restarted
May 11 15:11:19: %ATM-4-ILMICONFIGCHANGE: ILMI(ATM1/0/0): Restarting ATM signal.
May 11 15:11:19: ILMI(ATM1/0/0):Setting Local (Pre-Negotiated) PNNI version as d
May 11 15:11:19: ILMI(ATM1/0/0):Setting Local (Pre-Negotiated) UNI version as il
May 11 15:11:19: ILMI(ATM1/0/0):Registering New port
May 11 15:11:19: ILMI: Sending coldstart trap to peer
May 11 15:11:19: ILMI(ATM1/0/0): Sending ilmiColdStart trap
May 11 15:11:19: ILMI(ATM1/0/0): Sending ilmiColdStart trap
May 11 15:11:19: ILMI(ATM1/0/0): Querying peer device type.
```

ILMI 4.0 specifies only the coldStart trap and any enterprise–specific (i.e. vendor–specific) traps. ATM switches use the ilmiVccChange trap, as shown in the following sample output.

```
lwd: %LINK-3-UPDOWN: Interface ATM0/0/0, changed state to up
lwd: ILMI: Received Interface Up (ATM0/0/0)
lwd: ILMI(ATM0/0/0):Setting Local (Pre-Negotiated) PNNI version as ilmiPnniVersion1poi
lwd: ILMI(ATM0/0/0):Setting Local (Pre-Negotiated) UNI version as ilmiUniVersion4point
lwd: ILMI(ATM0/0/0):Registering New port
lwd: ILMI: Sending coldstart trap to peer
lwd: ILMI(ATM0/0/0): Sending ilmiColdStart trap (ATM0/0/0)
lwd: ILMI(ATM0/0/0): Sending ilmiVCCChange trap (ATM0/0/0)
lwd: ILMI(ATM0/0/0): Sending ilmiVCCChange trap (ATM0/0/0)
lwd: ILMI(ATM0/0/0): Sending ilmiColdStart trap
```

Use the `disable-ilmi-enterprise-traps` hidden command to disable ILMI enterprise traps.



**Caution:** Hidden commands are not officially supported by Cisco.

## Errored Responses

In some cases, the output of `debug atm ilmi` returns a message similar to the following:

```
*Sep 1 01:30:11: ILMI(ATM5/0): Errored response
Function Type = ilmiPeerDeviceInfo
```

By looking at this sample Sniffer trace, we can see that a standard SNMP header includes the following fields:

```
----- SNMP Header -----
SNMP: Version = 0
SNMP: Community = ILMI
SNMP: PDU = GetRequest
SNMP: Request identifier = 0x348 (840)
SNMP: Error status = noError (0)
SNMP: Error index = 0
```

The request ID is an integer that matches sent and received messages, and actually allows an ATM device to rapidly send several SNMP messages in a row, as we can see below.

The error–status field, when non–zero, indicates that an exception occurred while processing the request. The error–status field uses the following error values:

Value	Description
tooBig	The results of an operation would not fit in a single SNMP message.
noSuchName	Requested operation identified an unknown variable name, according to the community profile.
badValue	Requested operation specified an incorrect syntax or value when trying to modify a variable.
readOnly	Requested operation tried to modify a variable to which the community profile does not allow write-access.
genError	All other error conditions.

A non-zero value for the error index field indicates which variable in the request was in error. Non-zero values are possible only for the error values of noSuchName, badValue and readOnly.

## Sample Negotiation

Let's look at an example of the ILMI messages exchanged between two ATM interfaces.

During initialization and re-initialization, an ATM interface sends several GetRequest messages with different sequence numbers. The output of **debug snmp packet** reveals the unique contents of each GetRequest message. In the following sample output, ATM interface 0/0/0 sends six requests with sequence numbers from 6551 to 6556. Let's look at the GetRequests by breaking them down into two sets.

In the first set, ATM 0/0/0 sends the following two GetRequests:

Request ID	Action and Results
6551	Queries the atmfAtmLayerDeviceType object ID of the peer ATM interface. ATM end-systems take the value of user (1), while ATM network switches take the value of node (2).
6552	Queries the atmfAtmLayerUniType object ID of the peer ATM interface. Supported values are public and private.

```

1wld: ILMI(ATM0/0/0): Querying peer device type.
1wld: ILMI:peerDeviceTypeQuery not completed
1wld: ILMI:peerPortTypeQuery not completed
1wld: ILMI(ATM0/0/0): From Restarting To WaitDevAndPort
1wld: ILMI(ATM0/0/0):Sending out Request 6551
1wld: ILMI(ATM0/0/0):Sending out Request 6552
1wld: SNMP: Response, reqid 6551, errstat 0, erridx    0
atmfAtmLayerEntry.10.0 = 1
1wld: ILMI(ATM0/0/0):Response received for request    6551
1wld: SNMP: Response, reqid 6552, errstat 0, erridx    0
atmfAtmLayerEntry.8.0 = 2
1wld: ILMI(ATM0/0/0):Response received for request    6552
1wld: ILMI(ATM0/0/0): Peer Device Type is 1
1wld: The peer UNI Type on (ATM0/0/0) is 2
1wld: ILMI(ATM0/0/0): From WaitDevAndPort To DeviceAndPortComplete

```

lwld: ILMI(ATM0/0/0): From DeviceAndPortComplete To NodeConfigComplete

lwld: ILMI: My Device type is set to Node (ATM0/0/0)

In this second set of output, the switch sends five GetRequests. Each are listed in the table below. For ease of understanding, we have highlighted each series of messages in a different color below this table.

Request ID	Action and Results
6553	<p>Queries the atmfNetPrefixGroup object and implements the peerAddressTableCheck. We receive a GetResponse with an error. Matching the <b>debug snmp packet</b> output to the <b>debug atm ilmi</b> output, we see that the SetRequest queried an unknown variable, according to the community profile. The following output is also highlighted in <b>bold</b> below.</p> <pre>lwld: SNMP: Response, reqid 6553, errstat 2, erridx 1 atmfNetPrefixGroup.1 = NULL TYPE/VALUE lwld: ILMI(ATM0/0/0):Response received for request 6553 lwld: ILMI(ATM0/0/0): Errored response Function Type = ilmiAddressTableCheck</pre>
6554	<p>Queries three objects in the atmfAtmLayer table. Matching the <b>debug snmp packet</b> output to the <b>debug atm ilmi</b> output, we see that these objects are:</p> <ul style="list-style-type: none"><li>• Maximum number of active VPI bits</li><li>• Maximum number of active VCI bits</li><li>• UNI signaling version</li></ul> <p>The following output is also highlighted in blue below.</p> <pre>lwld: SNMP: Response, reqid 6554, errstat 0, erridx 0 atmfAtmLayerEntry.6.0 = 0 atmfAtmLayerEntry.7.0 = 10 atmfAtmLayerEntry.9.0 = 4 lwld: ILMI(ATM0/0/0): The Maximum # of VPI Bits is 0 lwld: ILMI(ATM0/0/0): The Maximum # of VCI Bits is 10 lwld: ILMI(ATM0/0/0): The UNI version is negotiated as ilmiUniVersion4point0</pre>
6555	<p>Queries five additional objects in the atmfAtmLayer table. Matching the debug snmp packet output to the debug atm ilmi output, we see that these objects are:</p> <ul style="list-style-type: none"><li>• Maximum number of VPCs</li><li>• Maximum number of VCCs</li><li>• Maximum SVPC VPI</li><li>• Maximum SVCC VPI</li><li>• Minimum SVCC VCI</li></ul> <p>The following output is also highlighted in <b>bold italics</b> below.</p> <pre>lwld: SNMP: Response, reqid 6555, errstat 0, erridx 0 atmfAtmLayerEntry.2.0 = 0 atmfAtmLayerEntry.3.0 = 4096 atmfAtmLayerEntry.13.0 = 0 atmfAtmLayerEntry.14.0 = 0 atmfAtmLayerEntry.15.0 = 0 lwld: ILMI(ATM0/0/0):Response received for request 6555</pre>

	<pre> lwd: ILMI(ATM0/0/0): Peer Max Vpcs is 0 lwd: ILMI(ATM0/0/0): Peer Max Vccs is 4096 lwd: ILMI(ATM0/0/0): Peer Max SvpcVpi is 0 lwd: ILMI(ATM0/0/0): Peer Max SvccVpi is 0 lwd: ILMI(ATM0/0/0): Peer Min SvccVci is 0 </pre>
6556	<p>Queries two objects in the physical port group:</p> <ul style="list-style-type: none"> <li>• atmfPortMyIfName</li> <li>• atmfMyIpNmAddress</li> </ul> <p>The following output is also highlighted in <i>italics</i> below.</p> <pre> lwd: SNMP: Response, reqid 6556, errstat 0, erridx 0   atmfPortEntry.7.0 = ATM2   atmfPhysicalGroup.2.0 = 10.10.10.4 lwd: ILMI(ATM0/0/0):Response received for request 6556  lwd: ILMI: The Neighbor's IfName on Intf (ATM0/0/0) is ATM2 lwd: ILMI: The Neighbor's IP on Intf (ATM0/0/0) is 168430084 </pre>
6557	<p>Sends a SetRequest with its network prefix and the far-end confirms validation and registration of this prefix. The following output is also highlighted in blue bold italics below.</p> <pre> lwd: ILMI(ATM0/0/0):Sending out Request 6557 lwd: SNMP: Response, reqid 6557, errstat 0, erridx 0   atmfNetPrefixEntry.3.0.13.71.0.145.129.0.0.0.0.96.62.90.143.1 = 1 lwd: ILMI(ATM0/0/0):Response received for request 6557 </pre>

```

lwd: ILMI(ATM0/0/0): Checking Peer Config and Address Table
lwd: ILMI:peerAddressTableCheck not completed
lwd: ILMI:peerConfigQuery not completed
lwd: ILMI:peerRangeConfigQuery not completed
lwd: ILMI(ATM0/0/0): From NodeConfigComplete To AwaitRestartAck

```

```

lwd: ILMI(ATM0/0/0):Sending out Request 6553
lwd: ILMI(ATM0/0/0):Sending out Request 6554
lwd: ILMI(ATM0/0/0):Sending out Request 6555
lwd: ILMI(ATM0/0/0):Sending out Request 6556

```

```

lwd: SNMP: Response, reqid 6553, errstat 2, erridx 1
  atmfNetPrefixGroup.1 = NULL TYPE/VALUE
lwd: ILMI(ATM0/0/0):Response received for request 6553
lwd: ILMI(ATM0/0/0): Errored response
Function Type = ilmiAddressTableCheck

```

```

lwd: SNMP: Response, reqid 6554, errstat 0, erridx 0
  atmfAtmLayerEntry.6.0 = 0
  atmfAtmLayerEntry.7.0 = 10
  atmfAtmLayerEntry.9.0 = 4
lwd: ILMI(ATM0/0/0):Response received for request 6554

```

```

lwd: SNMP: Response, reqid 6555, errstat 0, erridx 0
  atmfAtmLayerEntry.2.0 = 0
  atmfAtmLayerEntry.3.0 = 4096
  atmfAtmLayerEntry.13.0 = 0
  atmfAtmLayerEntry.14.0 = 0
  atmfAtmLayerEntry.15.0 = 0
lwd: ILMI(ATM0/0/0):Response received for request 6555

```

```

lwd: SNMP: Response, reqid 6556, errstat 0, erridx 0

```

```

atmfPortEntry.7.0 = ATM2
atmfPhysicalGroup.2.0 = 10.10.10.4
lwld: ILMI(ATM0/0/0):Response received for request 6556

lwld: ILMI(ATM0/0/0): The Maximum # of VPI Bits is 0
lwld: ILMI(ATM0/0/0): The Maximum # of VCI Bits is 10
lwld: ILMI(ATM0/0/0): The UNI version is negotiated as ilmiUniVersion4point0

lwld: ILMI(ATM0/0/0): Peer Max Vpcs is 0
lwld: ILMI(ATM0/0/0): Peer Max Vccs is 4096
lwld: ILMI(ATM0/0/0): Peer Max SvpcVpi is 0
lwld: ILMI(ATM0/0/0): Peer Max SvccVpi is 0
lwld: ILMI(ATM0/0/0): Peer Min SvccVci is 0

lwld: ILMI(ATM0/0/0): From AwaitRestartAck To UpAndNormal

lwld: ILMI: Auto Port determination enabled
lwld: ILMI(ATM0/0/0): Link determination completed
lwld: Peer Device Type: ilmiDeviceTypeUser
lwld: Peer Port Type: ilmiUniTypePrivate
lwld: Peer MaxVpiBits: 0
lwld: Peer MaxVciBits: 10
lwld: Peer MaxVpcs: 0
lwld: Peer MaxVccs: 4096
lwld: Peer MaxSvpcVpi: 0
lwld: Peer MaxSvccVpi: 0
lwld: Peer MinSvccVci: 0
lwld: Peer UNI version: ilmiUniVersion4point0
lwld: Neg. UNI Version: ilmiUniVersion4point0
lwld: Local Device Type: ilmiDeviceTypeNode
lwld: Local Port Type: ilmiPrivateUNINetworkSide
lwld: Local System ID:
lwld: ILMI(ATM0/0/0): KeepAlive disabled
lwld: ILMI: Sending Per-Switch prefix
lwld: ILMI: Registering prefix with end-system 47.0091.8100.0000.0060.3e5a.8f01

lwld: ILMI: The Neighbor's IfName on Intf (ATM0/0/0) is ATM2
lwld: ILMI: The Neighbor's IP on Intf (ATM0/0/0) is 168430084
lwld: ILMI(ATM0/0/0):Sending out Request 6557
lwld: SNMP: Response, reqid 6557, errstat 0, erridx 0

atmfNetPrefixEntry.3.0.13.71.0.145.129.0.0.0.0.96.62.90.143.1 = 1
lwld: ILMI(ATM0/0/0):Response received for request 6557

```

## ILMI Across NNI Interfaces

Network to network interfaces (NNI) define the connection between two ATM interfaces. In addition to all the UNI parameters described above, NNI ports negotiate the `atmfAtmLayernniSigVersion` object for the ATM layer group. This object indicates the latest version of the ATM Forum PNNI signaling specification that this ATM port supports. This object does not determine the PNNI routing version.

The values of the `atmfAtmLayerNniSigVersion` are:

- `iisp` (2)
- `pnniVersion1point0` (3)

**Note:** The UNI signaling version used on Interswitch Signaling Protocol (IISP) interfaces is determined by finding the highest common value advertised in the `atmfAtmLayerUniVersion` object. The interface type is

user-side if the local atmfMySystemIdentifier is larger than the peer's atmfMySystemIdentifier, and network-side if the local atmfMySystemIdentifier is smaller than the peer's atmfMySystemIdentifier.

**Note:** Although the IISP 1.0 specification states that IISP 1.0 links do not use ILMI, the ILMI 4.0 specification optionally specifies that ILMI functions other than address registration can run over IISP links.

---

## Related Information

- [ATM Technology Support Pages](#)
  - [Technical Support – Cisco Systems](#)
  - [More ATM Information](#)
- 

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2008 – 2009 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

---

Updated: Dec 18, 2007

Document ID: 10449

---