DSL Forum
Technical Report
TR-041

CORBA Specification
for
ADSL EMS-NMS Interface

June 2001

Abstract:
This Technical Report defines the first set of several information flows that enable automation of the DSL service business-to-business interfaces between various business entities involved in providing DSL service.

TABLE OF CONTENTS

1 INTRODUCTION

This document specifies the network management interface between EMS and NMS based on CORBA[1]. The object model is based on TR-035[2], which specifies the Protocol Independent Object Model for ADSL EMS-NMS Interface.

2 CORBA BASED NETWORK MANAGEMENT FRAMEWORK

This specification uses the CORBA management framework specified by ATM Forum's M4 Network View Interface CORBA Specification[4] as a basis. This ATM Forum specification specifies the CORBA information model for managing objects specified in the ATM M4 Network View[3], along with the CORBA services framework to support it. This specification extends that infrastructure to cover CORBA definitions for ADSL related objects specified in TR-035.

NOTE:
This version of the specification mainly covers configuration and performance management related functions for the ADSL EMS-NMS interface. Other functions will be addressed in a future revision.

3 CORBA MANAGEMENT INFORMATION MODEL

The CORBA management information model specified in this document is based on TR-035, which specifies the protocol independent object model to be used at the EMS-NMS interface for managing ADSL access subnetworks. A number of objects in TR-035 are adopted from standard object models specified by the ITU as well as the ATM Forum.

3.1 Access Subnetwork Managed Entities

Table 3-1 below lists the entities (managed object classes) specified in TR-035 for managing the ADSL access subnetwork, as well as shows the mapping of these to relevant CORBA IDL interface definitions.

This specification is based on the CORBA management framework specified by the ATM Forum in its M4 Network View Interface CORBA Specification [4]. The ATM Forum specification defines a general CORBA management framework, where all interfaces are derived from a common root interface named ManagedObject (MO). It also specifies CORBA IDL definitions for ATM related objects in TR-035 (marked as [M4] in Table 3-1 below).

CORBA IDL definitions for ADSL related objects are specified in this document.

Following objects presently do not have a standard CORBA specification under the ATM Forum's current CORBA framework. This framework is expected to evolve and align in the future with the CORBA framework being developed by the ITU.

- Element Management System (EMS)
- Network Element (managedElement)
- Equipment (equipment)
- Equipment Holder (equipment)
- Plug-in Unit (circuitPack)

Table 3-1. TR-035 to CORBA IDL Mapping Table

TR-035 ENTITY (OBJECT CLASS)	CORBA IDL INTERFACE
Access Subnetwork	[M4] Network
Layer Network Domain	[M4] AtmLND
Element Management System	-- NA --
Network Element	-- NA --
Equipment	-- NA --
Equipment Holder	-- NA --
Plug-in Unit	-- NA --
ADSL Line	ADSLLine
ADSL Channel	ADSLChannel
ADSL Configuration Profile	ADSLConfigurationProfile ADSLConfigurationProfileFac

TR-035 ENTITY (OBJECT CLASS)	CORBA IDL INTERFACE
	tory
ATM Subnetwork	[M4] AtmSubnetwork
ATM Subnetwork Connection	[M4] AtmSNC
ATM Link End	[M4] AtmLinkEnd
ATM Network CTP	[M4] AtmNetworkCTP
ATM Network TTP	[M4] AtmNetworkTTP
ATM Network Access Profile	[M4] AtmNetworkAccessProfile
ATM Traffic Descriptor	[M4] AtmTraficDesc

NOTE:

[M4] : These CORBA interfaces are specified by the ATM Forum[4].

-- NA -- : These CORBA interfaces are presently not standardized under the current ATM Forum CORBA framework.

TR-035 only covers configuration management related objects. This specification adds support for additional objects for performance management. Following IDL interfaces for performance management are specified in the ATM Forum CORBA framework[4]:

- CurrentData
- CurrentDataFactory
- CurrentBulkDataIterator
- HistoryData
- HistoryBulkDataIterator
- ThresholdData
- ThresholdBulkDataIterator
- BulkOperations
- PerformanceDataFileGenerator

This specification specifies IDL interface definitions for following additional objects that add performance management support for ADSL objects specified in TR-035. These interfaces are derived from the basic performance management interfaces listed above.

- ADSLLineCurrentDataAtuc
- ADSLLineHistoryDataAtuc
- ADSLLineCurrentDataAtur
- ADSLLineHistoryDataAtur
- ADSLChannelCurrentDataAtuc
- ADSLChannelHistoryDataAtuc
- ADSLChannelCurrentDataAtur
- ADSLChannelHistoryDataAtur

3.2 ADSL Configuration Management Interfaces

Each of the ADSL related managed object entity is modeled as a separate IDL interface. This section provides an overview of each IDL interface. Complete IDL definitions are available in Appendix-A.

3.2.1 ADSLLine

The ADSLLine IDL Interface implements the ADSL Line managed object as specified in TR-035. It inherits from the top-level managed object (ManagedObject Interface), and implements following methods:

- *getLineId* :
  Get the reference to an ADSL Line
- *getLineAll* :
  Get all parameters associated with an ADSL Line
- *getAssociatedChannels* :
  Get ADSL Channel(s) associated with an ADSL Line
- *getUnderlyingEquipmentUnits* :
  Get the equipment units associated with an ADSL Line
- *createChannel* :
  Configure a Channel associated with an ADSL Line
- *getAdministrativeState* :
  Get the Administrative State associated with an ADSL Line
- *setAdministrativeState* :
  Set the Administrative State associated with an ADSL Line
- *getOperationalState* :
  Get the Operational State associated with an ADSL Line
- *getInitFailureSwitch* :
  Get the state of the modem failure reporting flag associated with an ADSL Line
- *setInitFailureSwitch* :
  Set the state of the modem failure reporting flag associated with an ADSL Line
- *getCustomerId* :
  Get the Customer ID associated with an ADSL Line
- *setCustomerId* :
  Set the Customer ID associated with an ADSL Line
- *getLineAtucData* :
  Get the ATU-C Data associated with an ADSL Line
- *getLineAturData* :
  Get the ATU-R Data associated with an ADSL Line
- *getLineStatus* :
  Get the current Status of an ADSL Line

- *getConfigurationProfile* :
  Get the Configuration Profile associated with an ADSL Line
- *setConfigurationProfile* :
  Set the Configuration Profile associated with an ADSL Line
- *getCurrentOperationalMode*:
  Get the modem operational mode associated with an ADSL Line
- *getSupportedOperationalModes*:
  Get the modem operational modes supported by an ADSL Line
- *getAllowedOperationalModes*:
  Get the modem operational modes allowed for an ADSL Line
- *setAllowedOperationalModes*:
  Set the modem operational modes allowed for an ADSL Line

3.2.2 ADSLChannel

The ADSLChannel IDL Interface implements the ADSL Channel managed object as specified in TR-035. It inherits from the top-level managed object (ManagedObject Interface), and implements following methods:

- *getChannelId* :
  Get the reference to an ADSL Channel.
- *getChannelAll* :
  Get all parameters associated with an ADSL Channel.
- *getAssociatedLine* :
  Get the reference to the ADSL Line associated with an ADSL Channel.
- *getAdministrativeState* :
  Get the Administrative State associated with an ADSL Channel.
- *setAdministrativeState* :
  Set the Administrative State associated with an ADSL Channel.
- *getOperationalState* :
  Get the Operational State associated with an ADSL Channel.
- *getChannelType* :
  Get the Channel Type (latency) associated with an ADSL Channel.
- *getChannelAtucData* :
  Get the ATU-C data associated with an ADSL Channel.
- *getChannelAturData* :
  Get the ATU-R data associated with an ADSL Channel.

3.2.3 ADSLConfigurationProfile

The ADSLConfigurationProfile IDL Interface implements the ADSL Configuration Profile managed object as specified in TR-035. It inherits from the top-level managed object (ManagedObject Interface), and implements following methods:

- *getConfigurationProfileId* :
  Get the reference to an ADSL Configuration Profile
- *getConfigurationProfileAll* :
  Get all parameters associated with an ADSL Configuration Profile
- *getLineConfAtuc* :
  Get ATU-C Line configuration parameters
- *getLineConfAtur* :
  Get ATU-R Line configuration parameters
- *getFastChannelConfAtuc* :
  Get ATU-C Fast Channel configuration parameters
- *getFastChannelConfAtur* :
  Get ATU-R Fast Channel configuration parameters
- *getInterleavedChannelConfAtuc* :
  Get ATU-C Interleaved Channel configuration parameters
- *getInterleavedChannelConfAtur* :
  Get ATU-R Interleaved Channel configuration parameters

3.2.4 ADSLConfigurationProfileFactory

The ADSLConfigurationProfileFactory IDL Interface is used to create new instances of ADSL Configuration Profiles. It defines following methods:

- *getConfigurationProfileFactoryId* :
  Get a reference to the ADSL Configuration Profile Factory
- *createConfigurationProfile* :
  Create a new instance of an ADSL Configuration Profile

3.3 ADSL Performance Management Interfaces

This section provides an overview of IDL Interfaces for Performance Management operations for ADSL. The definitions are based on the ATM Forum's CORBA specification for M4 Network View[4]. Complete IDL definitions are available in Appendix-A.

3.3.1 ADSLLineCurrentDataAtuc

The ADSLLineCurrentDataAtuc IDL Interface is used to retrieve current performance data for the ADSL Line at the ATU-C. It is derived from the corba_pm::CurrentData interface, and defines no additional methods.

3.3.2 ADSLLineHistoryDataAtuc

The ADSLLineHistoryDataAtuc IDL Interface is used to retrieve historic (past) performance data for the ADSL Line at the ATU-C. It is derived from the corba_pm::HistoryData interface, and defines no additional methods.

3.3.3 ADSLLineCurrentDataAtur

The ADSLLineCurrentDataAtur IDL Interface is used to retrieve current performance data for the ADSL Line at the ATU-R. It is derived from the corba_pm::CurrentData interface, and defines no additional methods.

3.3.4 ADSLLineHistoryDataAtur

The ADSLLineHistoryDataAtur IDL Interface is used to retrieve historic performance data for the ADSL Line at the ATU-R. It is derived from the corba_pm::HistoryData interface, and defines no additional methods.

3.3.5 ADSLChannelCurrentDataAtuc

The ADSLChannelCurrentDataAtuc IDL Interface is used to retrieve current performance data for an ADSL Channel at the ATU-C. It is derived from the corba_pm::CurrentData interface, and defines no additional methods.

3.3.6 ADSLChannelHistoryDataAtuc

The ADSLChannelHistoryDataAtuc IDL Interface is used to retrieve historic performance data for an ADSL Channel at the ATU-C. It is derived from the corba_pm::HistoryData interface, and defines no additional methods.

3.3.7 ADSLChannelCurrentDataAtur

The ADSLChannelCurrentDataAtur IDL Interface is used to retrieve current performance data for an ADSL Channel at the ATU-R. It is derived from the corba_pm::CurrentData interface, and defines no additional methods.

3.3.8 ADSLChannelHistoryDataAtur

The ADSLChannelHistoryDataAtur IDL Interface is used to retrieve historic performance data for an ADSL Channel at the ATU-R. It is derived from the corba_pm::HistoryData interface, and defines no additional methods.

4 REFERENCES

[1] The Object Management Group (OMG), "The Common Object Request Broker: Architecture and Specification", OMG Document: formal/99-10-07, Revision 2.3.1, October 1999.

[2] DSL Forum (DSLF), "Protocol Independent Object Model for ADSL EMS-NMS Interface.", DSLF Technical Report: TR-035, March 2000.

[3] The ATM Forum (ATMF), "M4 Interface Requirements and Logical MIB: ATM Network View", version 2, ATMF Specification: af-nm-0058.001, May 1999.

[4] The ATM Forum (ATMF), "CORBA Specification for M4 Interface: Network View", ATMF Letter Ballot Document: fb-nm-0166.000, Apr 2001.

[5] DSL Forum (DSLF), "Proposed IDL definitions for Performance Management portion of WT-046v1", Alcatel (R. Abbi & A. Tuzel), DSLF Contribution : dslforum2000.113, May 2000.

[6] DSL Forum (DSLF), "Proposed updates to the Performance Management CORBA IDL definitions", Telcordia Technologies (A. Mayer, B. Atwater, K. Armington), DSLF Contribution: dslforum2000.258, August 2000.

[7] DSL Forum (DSLF), "Update of WT-046v3 IDL definitions for Performance Management", Alcatel (R. Abbi & A. Tuzel) and Telcordia Technologies (A. Mayer & K. Armington), DSLF Contribution: dslforum2000.428, December 2000.

Appendix A : IDL For ADSL Management

A.1 : ADSL Configuration Management Module (dslf_adsl.idl)

```
#ifndef _dslf_adsl_idl_
#define _dslf_adsl_idl_

#include "NetMgmt.idl"

/*****************************************************************************/
/*  This module defines the interfaces for managing ADSL transmission lines. */
/*  It is based on the CORBA Network Management Framework defined in the     */
/*  NetMgmt module in the ATM Forum's CORBA Specification for M4 Interface:  */
/*  NetworkView specification (fb-nm-0166.000)                               */
/*  Interfaces defined in this module are based on the ADSL Management       */
/*  Object Model specified in TR-035.                                        */
/*****************************************************************************/
module dslf_adsl {

const string moduleName = "dslf_adsl";

// Interfaces imported from NetMgmt
// --------------------------------
#define MOID                                NetMgmt::MOID;
#define MOIDList                            NetMgmt::MOIDList;
#define Name                                NetMgmt::NameType;
#define AdministrativeState                 NetMgmt::AdministrativeState;
#define OperationalState                    NetMgmt::OperationalState;
#define ManagedObject                       NetMgmt::ManagedObject;
#define ManagedObjectFactory                NetMgmt::ManagedObjectFactory;
#define Portal                              NetMgmt::Portal;

// Exceptions imported from NetMgmt
// --------------------------------
#define DuplicateItem                       NetMgmt::DuplicateItem;
#define DuplicateName                       NetMgmt::DuplicateName;
#define ItemNotFound                        NetMgmt::ItemNotFound;
#define NotSupported                        NetMgmt::NotSupported;
#define ObjectFailure                       NetMgmt::ObjectFailure;
#define UID                                 NetMgmt::UID;


// Interfaces defined in this module
// --------------------------------
interface ADSLChannel;
interface ADSLLine;
interface ADSLConfigurationProfile;
interface ADSLConfigurationProfileFactory;
```

```
// Data Structure Definitions
// --------------------------

/** Reference to an ADSL Configuration Profile */
struct ADSLConfigurationProfileId {
    ADSLConfigurationProfile        ref;
    Name                            name;
};

/** Reference to an ADSL Configuration Profile Factory */
struct ADSLConfigurationProfileFactoryId {
    ADSLConfigurationProfileFactory  ref;
    Name                            name;
};

/** Reference to an ADSL Channel */
struct ADSLChannelId {
    ADSLChannel                     ref;
    Name                            name;
};

/** List of ADSL Channel references */
typedef sequence <ADSLChannelId> ADSLChannelList;

/** Reference to an ADSL Line */
struct ADSLLineId {
    ADSLLine                        ref;
    Name                            name;
};

/** Options for ADSL line codes */
enum LineCoding {
    dmt,
    cap,
    qam,
    other
};

/** Options for ADSL Channel latency */
enum ChannelType {
    noChannel,
    fastOnly,
    interleavedOnly,
    fastAndInterleaved,
    fastOrInterleaved
};

/** Options for ADSL line rate mode */
enum RateMode {
    fixed,
    adaptAtStartup,
    adaptAtRuntime
};
```

```
/** Options for modem operating modes */
enum OperationalMode {
    ansi,
    etsi,
    potsNonOverlapped,
    potsOverlapped,
    isdnNonOverlapped,
    isdnOverlapped,
    isdnTcm,
    potsNonOverlappedLite,
    potsOverlappedLite,
    isdnTcmLite
};

/** List of Operational Modes */
typedef sequence <OperationalMode> OperationalModes;

/** Options for ADSL line conditions */
enum LineCondition {
    atuc_lof,
    atuc_los,
    atuc_lpr,
    atuc_lol,
    atuc_lossOfSignalQuality,
    atuc_dataInitfailure,
    atuc_configInitFailure,
    atuc_protocolInitFailure,
    atuc_noPeerPresent,
    atuc_bitRateThreshold,
    atur_lof,
    atur_los,
    atur_lpr,
    atur_lossOfSignalQuality,
    atur_bitRateThreshold
};

/** List of ADSL line conditions */
typedef sequence <LineCondition> LineConditions;

/** ADSL Line related operational data */
struct LineData {

    /** This read-only attribute provides the Line Coding information */
    LineCoding lineCoding;

    /** This read-only attribute provides the Current signal/noise margin
       for the received signal in 1/10th of a dB.*/
    long currentSNRMargin;

    /** This read-only attribute indicates the measured difference in
       total power transmitted by the peer ATU and the total power
       received by this ATU in 1/10th of a dB */
    long currentAttenuation;

    /** This read-only attribute indicates the measured power transmitted
       by the ATU in 1/10th of a dB */
    long currentOutputPower;

    /** This read-only attribute indicates the maximum attainable rate
       in kbps */
    long currentAttainableRate;
```

```
    /** This read-only attribute indicates the current line rate in kbps */
    long currentLineRate;

    /** This read-only attribute indicates the previous line rate in kbps */
    long previousLineRate;

    /** This attribute indicates the supported Channel types over this
       ADSL line */
    ChannelType supportedChannelTypes;

};



/** All ADSL Line parameters */
struct ADSLLineAll {
    ADSLLineId                 lineId;
    ADSLChannelList            associatedChannels;
    MOIDList                   underlyingEquipmentUnits;
    AdministrativeState        adminState;
    OperationalState           operState;
    boolean                    initFailureSwitch;
    string                     customerId;
    LineData                   lineAtucData;
    LineData                   lineAturData;
    LineConditions             lineStatus;
    ADSLConfigurationProfileId configurationProfile;
    OperationalMode            currentOperationalMode;
    OperationalModes           allowedOperationalModes;
    OperationalModes           supportedOperationalModes;
};


/** ADSL Channel operating data */
struct ChannelData {

    /** This read-only attribute indicates the current Channel rate in kbps */
    long currentChannelRate;

    /** This read-only attribute indicates the previous Channel rate in kbps */
    long previousChannelRate;

    /**This attribute indicates the current interleave delay in mili-seconds */
    long interleaveDelay;  /* For Interleaved Channel Only */

    /** This attribute indicates the current length, in bytes, of the
       Channel data-block on which CRC is calculated */
    long crcBlockLength;
};
```

```
/** All ADSL Channel data */
struct ADSLChannelAll {
    ADSLChannelId       ChannelId;
    ADSLLineId          associatedLineId;
    AdministrativeState adminState;
    OperationalState    operState;
    ChannelType         ChannelType;
    ChannelData         atucData;
    ChannelData         aturData;
};

/** ADSL Line configuration parameters */
struct LineConf {

    /** This attribute configures the modem rate adaptation mode */
    RateMode rateMode;

    /** This attribute configures the target signal/noise margin, in 1/10th
       of dB, the modem must achieve with BER of 10-7 or better */
    long targetSNRMargin;

    /** This attribute configures the maximum signal/noise margin, in 1/10th
       of dB, the modem must try to maintain before increasing the data rate */
    long maxSNRMargin;

    /** This attribute configures the minimum acceptable signal/noise margin,
       in 1/10th of dB */
    long minSNRMargin;
};

/** ADSL Channel configuration parameters */
struct ChannelConf {

    /** This attribute configures the minimum acceptable transmission rate,
       for the Channel in bps */
    long minTxRate;

    /** This attribute configures the maximum allowed transmission rate,
       for the Channel in bps */
    long maxTxRate;

    /** This attribute configures the maximum allowed interleave delay,
       in miliseconds, for the interleaved Channel */
    long maxInterleaveDelay; /* For Interleaved Channel Only */
};

/** All ADSL Configuration Profile parameters */
struct ADSLConfigurationProfileAll {
    ADSLConfigurationProfileId profileId;
    string                     profileName;
    LineConf                   lineConfAtuc;
    LineConf                   lineConfAtur;
    ChannelConf                fastChannelConfAtuc;
    ChannelConf                fastChannelConfAtur;
    ChannelConf                interleavedChannelConfAtuc;
    ChannelConf                interleavedChannelConfAtur;
};
```

```
/**-------------------------------------------------------------------
 *         ADSL Line Interface
 *-------------------------------------------------------------------
 * This interface represents an ADSL Line.
 *-------------------------------------------------------------------
 */
interface ADSLLine : ManagedObject, Portal {

    /** Get the reference to an ADSL Line */
    ADSLLineId getLineId (in ADSLLineId lineId)
        raises (ObjectFailure);

    /** Get all parameters associated with an ADSL Line */
    ADSLLineAll getLineAll (in ADSLLineId lineId)
        raises (ObjectFailure, NotSupported);

    /** Get the Channels associated with an ADSL Line */
    ADSLChannelList getAssociatedChannels (in ADSLLineId lineId)
        raises (ObjectFailure);

    /** Get the equipment unit(s) supporting an ADSL Line */
    MOIDList   getUnderlyingEquipmentUnits (in ADSLLineId lineId)
        raises (ObjectFailure);

    /** Configure a Channel for an ADSL Line */
    void createChannel     (in  ADSLLineId    lineId,
                            in  ChannelType   ChannelType,
                            out ADSLChannelId ChannelId)
        raises (ObjectFailure, NotSupported);

    /** Get the Administrative State of an ADSL Line */
    AdministrativeState getAdministrativeState (in ADSLLineId lineId)
        raises (ObjectFailure);

    /** Set the Administrative State of an ADSL Line */
    void setAdministrativeState (in ADSLLineId          lineId,
                                 in AdministrativeState adminState)
        raises (ObjectFailure);

    /** Get the Operational State of an ADSL Line */
    OperationalState getOperationalState (in ADSLLineId lineId)
        raises (ObjectFailure);

    /** Get modem initialization failure notification flag.
        Initialization failure alarms are reported if 'True' */
    boolean getInitFailureSwitch (in ADSLLineId lineId)
        raises (ObjectFailure, NotSupported);

    /** Set modem initialization failure notification flag.
        Initialization failure alarms are reported if 'True' */
    void setInitFailureSwitch (in boolean initFailureSwitch)
        raises (ObjectFailure, NotSupported);

    /** Get the Customer ID associated with an ADSL Line */
    string getCustomerID (in ADSLLineId lineId)
        raises (ObjectFailure);

    /** Set the Customer ID associated with an ADSL Line */
    void setCustomerID (in ADSLLineId lineId, in string customerId)
        raises (ObjectFailure);
```

```
    /** Get operating data for the ATU-C end of an ADSL Line */
    LineData getLineAtucData (in ADSLLineId lineId)
        raises (ObjectFailure);

    /** Get operating data for the ATU-R end of an ADSL Line */
    LineData getLineAturData (in ADSLLineId lineId)
        raises (ObjectFailure);

    /** Get the Status of an ADSL Line */
    LineConditions getLineStatus (in ADSLLineId lineId)
        raises (ObjectFailure);

    /** Get the ADSL Configuration Profile associated with an ADSL Line */
    ADSLConfigurationProfileId getADSLConfigurationProfile (
                                                in ADSLLineId lineId)
        raises (ObjectFailure, NotSupported);

    /** Assign a new Configuration Profile to an ADSL Line */
    void setADSLConfigurationProfile (in ADSLLineId lineId,
                                      in ADSLConfigurationProfile confProfile )
        raises (ObjectFailure, NotSupported, InvalidReference);

    /** Get current modem operational mode */
    OperationalMode getCurrentOperationalMode (in ADSLLineId lineId)
        raises (ObjectFailure);

    /** Get operational modes supported by an ADSL Line */
    OperationalModes getSupportedOperationalModes (in ADSLLineId lineId)
        raises (ObjectFailure);

    /** Get operational modes allowed for an ADSL Line */
    OperationalModes getAllowedOperationalModes (in ADSLLineId lineId)
        raises (ObjectFailure);

    /** Set operational modes allowed for an ADSL Line */
    void setAllowedOperationalModes (in ADSLLineId lineId,
                                     in OperationalModes allowedModes)
        raises (ObjectFailure, NotSupported);

}; // end of ADSLLine interface


/**-------------------------------------------------------------------
 *          ADSL Channel Interface
 *-------------------------------------------------------------------
 * This interface represents an ADSL Channel. Both ends of the channel
 * (ATU-C and ATU-R) are represented by this interface.
 *-------------------------------------------------------------------
 */
interface ADSLChannel : ManagedObject, Portal {

    /** Get the reference to an ADSL Channel */
      ADSLChannelId getADSLChannelId (in ADSLChannelId ChannelId)
      raises (ObjectFailure); // returns the name of the object

    /** Get all parameters associated with an ADSL Channel */
    ADSLChannelAll getADSLChannelAll (in ADSLChannelId ChannelId)
        raises (ObjectFailure, NotSupported);
```

```
    /** Get the reference to the ADSL Line associated with an ADSL Channel */
    ADSLLineId getAssociatedADSLLineID (in ADSLChannelId ChannelId)
        raises (ObjectFailure);

    /** Get the Administrative State of an ADSL Channel */
    AdministrativeState getAdministrativeState (in ADSLChannelId ChannelId)
        raises (ObjectFailure);

    /** Set the Administrative State of an ADSL Channel */
    void setAdministrativeState (in ADSLChannelId ChannelId ,
                                 in AdministrativeState adminState)
        raises (ObjectFailure);

    /** Get the Operational State of an ADSL Channel */
    OperationalState getOperationalState (in ADSLChannelId ChannelId)
        raises (ObjectFailure);

    /** Get the Channel type (latency) of an ADSL Channel */
    ChannelType getChannelType (in ADSLChannelId ChannelId)
        raises (ObjectFailure);

    /** Get data for the ATU-C end of an ADSL Channel */
    ChannelData getChannelAtucData (in ADSLChannelId ChannelId)
        raises (ObjectFailure);

    /** Get data for the ATU-R end of an ADSL Channel */
    ChannelData getChannelAturData (in ADSLChannelId ChannelId)
        raises (ObjectFailure);

}; // end of ADSLChannel interface


/**-------------------------------------------------------------------
 *         ADSL Configuration Profile Interface
 *-------------------------------------------------------------------
 * This interface represents an ADSL Configuration Profile.
 * Configuration Profiles are created via the
 * ADSLConfigurationProfileFactory Interface.
 *-------------------------------------------------------------------
 */
interface ADSLConfigurationProfile : ManagedObject, Portal {

    /** Get the reference to an ADSL Configuration Profile */
    ADSLConfigurationProfileId getADSLConfigurationProfileId (in
                                          ADSLConfigurationProfileId id)
        raises (ObjectFailure);

    /** Get all parameters associated with a Configuration Profile */
    ADSLConfigurationProfileAll getADSLConfigurationProfileAll (in
                                          ADSLConfigurationProfileId id)
        raises (ObjectFailure, NotSupported);

    /** Get ATU-C configuration data associated with a Configuration Profile */
    LineConf getLineConfAtuc (in ADSLConfigurationProfileId id)
        raises (ObjectFailure, NotSupported);

    /** Get ATU-R configuration data associated with a Configuration Profile */
    LineConf getLineConfAtur (in ADSLConfigurationProfileId id)
        raises (ObjectFailure, NotSupported);
```

```
    /** Get ATU-C Fast Channel configuration data associated with a
        Configuration Profile */
    ChannelConf getFastChannelConfAtuc (in ADSLConfigurationProfileId id)
        raises (ObjectFailure, NotSupported);

    /** Get ATU-R Fast Channel configuration data associated with a
        Configuration Profile */
    ChannelConf getFastChannelConfAtur (in ADSLConfigurationProfileId id)
        raises (ObjectFailure, NotSupported);

    /** Get ATU-C Interleaved Channel configuration data associated with a
        Configuration Profile */
    ChannelConf getInterleavedChannelConfAtuc (
                                        in ADSLConfigurationProfileId id)
        raises (ObjectFailure, NotSupported);

    /** Get ATU-R Interleaved Channel configuration data associated with a
        Configuration Profile */
    ChannelConf getInterleavedChannelConfAtur (
                                        in ADSLConfigurationProfileId id)
        raises (ObjectFailure, NotSupported);

}; // end of ADSLConfigurationProfile interface


/**-------------------------------------------------------------------
 *         ADSL Configuration Profile Factory Interface
 *-------------------------------------------------------------------
 * This interface is used to create new instances of ADSL Configuration
 * Profiles.
 *-------------------------------------------------------------------
 */
interface ADSLConfigurationProfileFactory : ManagedObject  {

    /** Get the reference to the Configuration Profile Factory */
    ADSLConfigurationProfileFactoryId getADSLConfigurationProfileFactoryId(
                                    in ADSLConfigurationProfileFactoryId id)
        raises (ObjectFailure);

    /** Create a new instance of an ADSL Configuration Profile */
    ADSLConfigurationProfileId createADSLConfigurationProfile(
                        in ADSLConfigurationProfileFactoryId id,
                        in MOID          containerObject,
                        in LineConf      lineConfAtuc,
                        in LineConf      lineConfAtur,
                        in ChannelConf   fastChannelConfAtuc,
                        in ChannelConf   fastChannelConfAtur,
                        in ChannelConf   interleavedChannelConfAtuc,
                        in ChannelConf   interleavedChannelConfAtur)
        raises ( ObjectFailure, NotSupported, OutOfRange, ItemNotFound);

}; // end of ADSLConfigurationProfileFactory


}; // end of dslf_adsl module
/*************************************************************************/

#endif // end of ifndef _dslf_adsl_idl_
```

A.2: ADSL Performance Management Module (dslf_adsl_pm.idl)

```
#ifndef _dslf_adsl_pm_idl_
#define _dslf_adsl_pm_idl_

#include "corba_pm.idl"

/*************************************************************************/
/* This module defines the interfaces for supporting Performance Management */
/* for ADSL transmission lines.                                          */
/* It is based on the CORBA Network Management Framework defined in the  */
/* corba_pm module in the ATM Forum's CORBA Specification for M4 Interface: */
/* NetworkView specification (fb-nm-0166.000)                            */
/*************************************************************************/

module dslf_adsl_pm {

const string moduleName = "dslf_adsl_pm";

// Interfaces imported from corba_pm
// ---------------------------------
#define   CurrentData         corba_pm::CurrentData;
#define   HistoryData         corba_pm::HistoryData;

/* Forward declarations */
/* ==================== */
interface ADSLLineCurrentDataAtuc;
interface ADSLLineHistoryDataAtuc;
interface ADSLLineCurrentDataAtur;
interface ADSLLineHistoryDataAtur;
interface ADSLChannelCurrentDataAtuc;
interface ADSLChannelHistoryDataAtuc;
interface ADSLChannelCurrentDataAtur;
interface ADSLChannelHistoryDataAtur;


/* Type Definitions */
/* ================ */

/** ADSL Performance Data Set (PerfDataSet) Values
    ------------------------------------------- */
const short adslLineAtuc      = 1;
const short adslLineAtur      = 2;
const short adslChannelAtuc   = 3;
const short adslChannelAtur   = 4;
```

```
/** ADSL Performance Counter (PerfParameter) Values
    ------------------------------------------- */
/** ADSL Line Counters */
const short lossOfFrameEvents             = 1;
const short lossOfSignalEvents            = 2;
const short lossOfLinkEvents              = 3;
const short lossOfPowerEvents             = 4;
const short erroredSeconds                = 5;
const short severelyErroredSeconds        = 6;
const short unavailableSeconds            = 7;
const short initializationEvents          = 8;
const short fastRetrainAttempts           = 9;
const short failedFastRetrainAttempts     = 10;
const short CodeViolations                = 11;
const short ForwardErrorCorrections       = 12;
const short ForwardErrorCorrectionSeconds = 13;
const short LossOfSignalSeconds           = 14;

/** ADSL Channel Counters */
const short transmittedBlocks             = 100;
const short receivedBlocks                = 101;
const short correctedBlocks               = 102;
const short uncorrectedBlocks             = 103;

/* Interfaces and Methods */
/* ====================== */

/** ADSLLineCurrentDataAtuc interface
    -------------------------------- */
interface ADSLLineCurrentDataAtuc : CurrentData {
      /** This interface implements the CurrentData for
       * ADSL Line related counters for ATU-C.
       * It does not define any additional methods. */
};

/** ADSLLineHistoryDataAtuc interface
    -------------------------------- */
interface ADSLLineHistoryDataAtuc : HistoryData {
      /** This interface implements the HistoryData for
       * ADSL Line related counters for ATU-C.
       * It does not define any additional methods. */
};

/** ADSLLineCurrentDataAtur interface
    -------------------------------- */
interface ADSLLineCurrentDataAtur : CurrentData {
      /** This interface implements the CurrentData for
       * ADSL Line related counters for ATU-R.
       * It does not define any additional methods. */
};

/** ADSLLineHistoryDataAtur interface
 *  -------------------------------- */
interface ADSLLineHistoryDataAtur : HistoryData {
      /** This interface implements the HistoryData for
       * ADSL Line related counters for ATU-R.
       * It does not define any additional methods. */
};
```

```
/** ADSLChannelCurrentDataAtuc interface
    ---------------------------------- */
interface ADSLChannelCurrentDataAtuc : CurrentData {
      /** This interface implements the CurrentData for
       * ADSL Channel related counters for ATU-C.
       * It does not define any additional methods. */
};

/** ADSLChannelHistoryDataAtuc interface
    ---------------------------------- */
interface ADSLChannelHistoryDataAtuc : HistoryData {
      /** This interface implements the HistoryData for
       * ADSL Channel related counters for ATU-C.
       * It does not define any additional methods. */
};

/** ADSLChannelCurrentDataAtur interface
    ---------------------------------- */
interface ADSLChannelCurrentDataAtur : CurrentData {
      /** This interface implements the CurrentData for
       * ADSL Channel related counters for ATU-R.
       * It does not define any additional methods. */
};

/** ADSLChannelHistoryDataAtur interface
 *  ---------------------------------- */
interface ADSLChannelHistoryDataAtur : HistoryData {
      /** This interface implements the HistoryData for
       * ADSL Channel related counters for ATU-R.
       * It does not define any additional methods. */
};


}; // end of dslf_adsl_pm module

/*****************************************************************************/
#endif // end of ifndef _dslf_adsl_pm_
```