The ATM Forum

Technical Committee Network Management

M4 Interface: ATM Network View CORBA MIB Version 2

AF-NM-0185.000

August, 2002

The ATM Forum
Worldwide Headquarters
2570 West El Camino Real, Suite 304
Mountain View, CA 94040-1313
Tel:+1-650-949-6700
Fax:+1-650-949-6705

Editors:

Weijing Chen
Andrew J. Mayer, Ph.D.

Table of Contents

1. Background

ITU-T Recommendation Q.816 defines the framework support services that cover CORBA based TMN services including naming service, notification service, telecom log service, messaging service, security service, transaction service, factory finder service, channel finder service, terminator service, MOO service and heartbeat service. ITU-T Recommendation X.780 defines the framework modeling guidelines and super-classes for all managed objects, standard set of notifications, and managed object factories. Recommendation X.780 and Recommendation Q.816 together form the CORBA-based TMN interface framework.

ITU-T framework originally dictated that each managed object had to be an individually-accessible CORBA object, often known as a “fine-grained” approach. Recently, however, it has been extended to enable implementers to instead choose an approach where managed objects do not have individual object references. This “coarse-grained” approach relieves implementers from having to support large numbers of CORBA interoperable object references (IORs).

Recommendation Q.816.1 is an extension to Recommendation Q.816, and enables the framework to support "coarse-grained" interfaces. Recommendation Q.816.1 adds one new service to be supported on the interface. Like Recommendation Q.816.1, Recommendation X.780.1 defines extensions to Recommendation X.780 for coarse-grained interfaces. Recommendation X.780 defines rules for modeling network resources on CORBA interfaces in a "fine-grained" manner. X.780.1 does the same thing, but in a "coarse-grained" manner. Semantically, the fine-grained and coarse-grained approaches provide access to practically identical information and operations; they simply differ in the syntax used to access the data and capabilities. Rather than interacting directly with object representing manageable resources through their CORBA interfaces, the interactions are channeled through a "façade" interface. This approach has the benefit of reducing the number of CORBA objects with which a management system must interact, reducing overhead in large systems.

Recommendation M.3120 defines a generic information model and provides a CORBA IDL version of the generic network information model originally defined in Recommendation M.3100, its amendments and G.855.1. Some of the object interfaces defined in it are suitable for direct use, while others are intended to be specialized for use with a particular network technology. The idea is that these objects form a core set of capabilities that may then be extended to meet the needs of any particular technology. These extensions are expected to be defined mainly by industry forums and standards groups in addition to the ITU-T. Recommendation M.3120 contains both fine and coarse-grained interface definitions for this generic information model.

Current ATM Forum CORBA model is based on early working version of ITU-T CORBA framework. It is in the best interest of both service provider and equipment supplier to have one CORBA framework for different network technologies. Also ATM Forum can focus its effort on ATM network modeling and have worldwide CORBA experts working on basic framework issues much like CMIP/GDMO eras.

2. Introduction

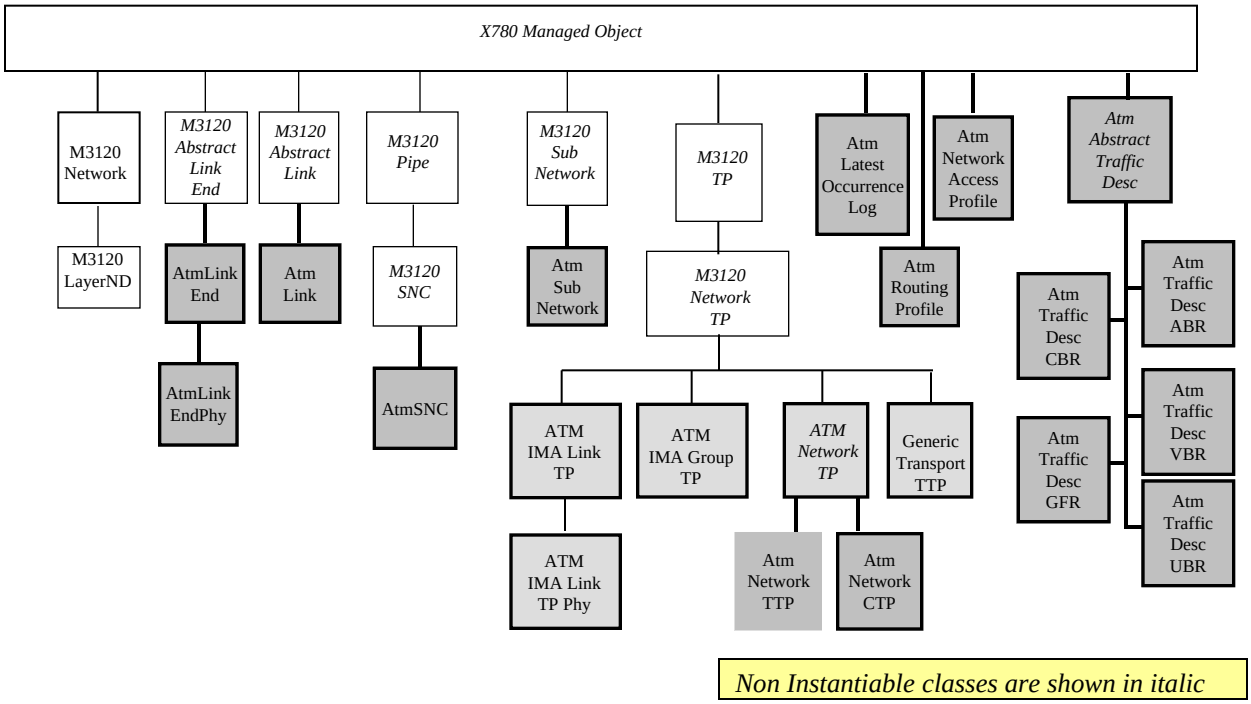
This document specifies a CORBA-based ATM network management interface definition that provides a formal representation of the information exchanged between a managed system and a managing system. The interface specification was defined specifically to meet the criteria set forth by ATM Forum "M4 Interface Requirements and Logical MIB: ATM Network View" version 2 (af-nm-0058.001)[1]. It also includes necessary managed objects from ATM Forum "M4 Interface Requirements and Logical MIB: ATM Network Element View" (af-nm-0020.001)[2] and other ATM SVC related managed objects.

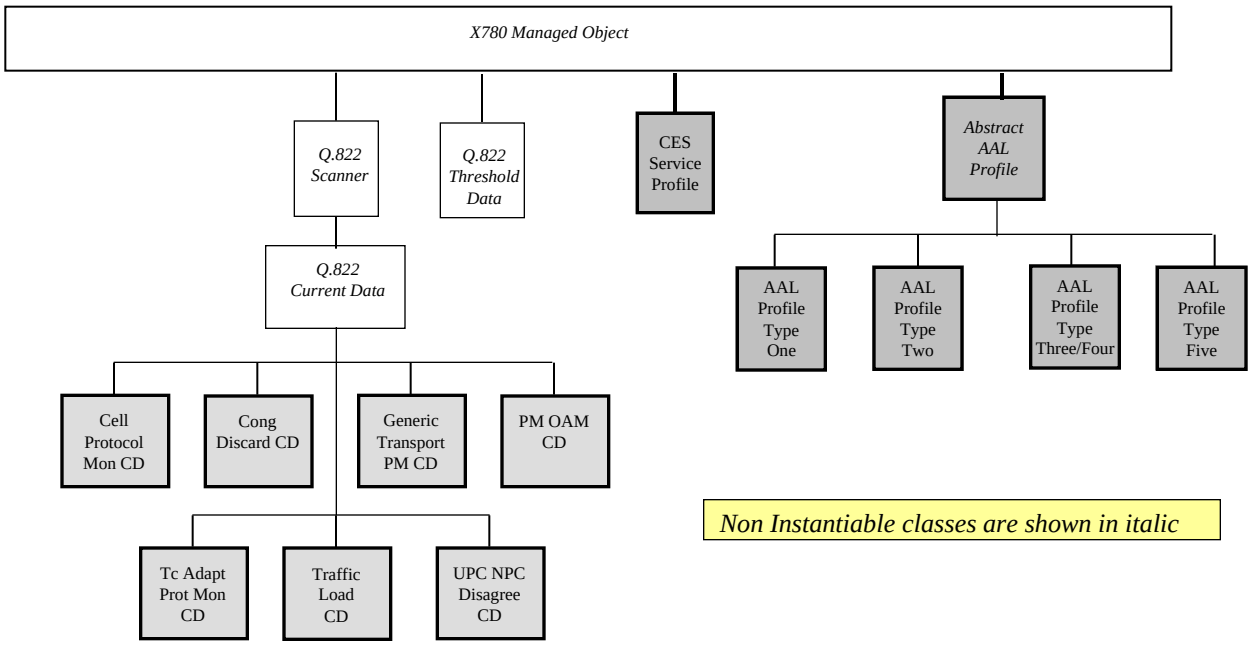
This interface definition is fully compliant with ITU-T Recommendation X.780.1 "TMN Guidelines for Defining Coarse-grained CORBA Managed Objects" [4] and uses Recommendation Q.816.1 "CORBA Based TMN Services Extensions to Support Coarse-Grained Interfaces" [6] and Recommendation Q.822.1 “CORBA-Based TMN Performance Management Service”[8]. The managed objects defined are inherited from M.3120 "CORBA Generic Network and NE Level Information Model" [7] whenever possible.

The CORBA IDL contained in this document was checked with VisiBroker for Java 4.5 and ORBacus HIDL 4.0.5. Use of these tools does not constitute an endorsement of these tools by the ATM Forum or ATM Forum member companies.

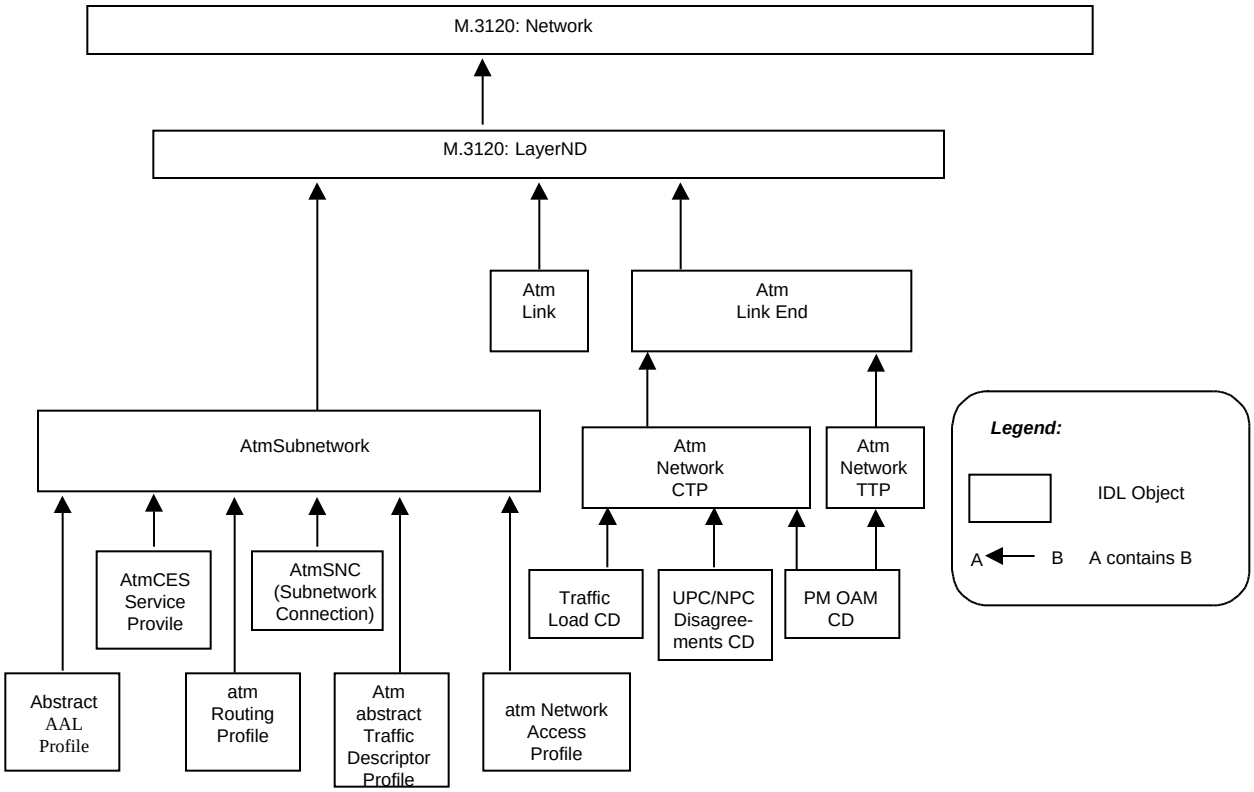
3. Overview of the Information Model

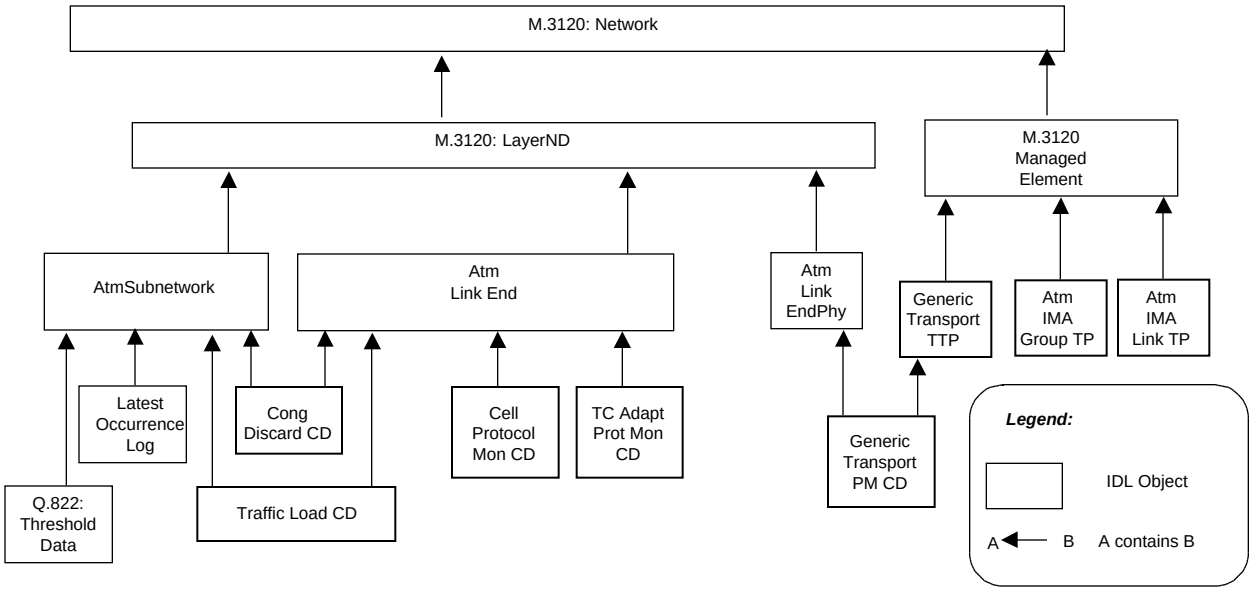
3.1. *Interface Inheritance Relationship*





3.2. *Interface Containment Relationship*





3.3. *Object Dependancy Relationships*

Subordinate	Superior	superior Subclasses Allowed	subordinate Subclasses Allowed	manager Creates Allowed	DeletePolicy
AtmLink	LayerND	True	True	True	itut_x780::deleteOnlyIfNoContainedObjects
AbstractAalProfile	AtmSubnetwork	True	True	True	itut_x780::deleteOnlyIfNoContainedObjects
AtmImaGroup	ManagedElement	True	True	True	itut_x780::deleteOnlyIfNoContainedObjects
AtmImaLink	ManagedElement	True	True	True	itut_x780::deleteOnlyIfNoContainedObjects
AtmLinkEnd	LayerND	True	True	True	itut_x780::deleteOnlyIfNoContainedObjects
AtmLinkEndPhy	LayerND	True	True	False	itut_x780::notDeletable
LayerND	Network	True	True	False [1]	itut_x780::notDeletable
AtmNetworkAccessProfile	AtmSubnetwork	True	True	True	itut_x780::deleteOnlyIfNoContainedObjects
AtmNetworkCTP	AtmLinkEnd	True	True	True	itut_x780::deleteOnlyIfNoContainedObjects
AtmNetworkTTP	AtmLinkEnd	True	True	True	itut_x780::deleteOnlyIfNoContainedObjects
AtmRoutingProfile	AtmSubnetwork	True	True	True	itut_x780::deleteOnlyIfNoContainedObjects
AtmSNC	AtmSubnetwork	True	True	True	itut_x780:: notDeletable
AtmSubnetwork	LayerND	True	True	False	itut_x780::notDeletable
AtmAbstractTrafficDescProfile	AtmSubnetwork	True	True	True	itut_x780::deleteOnlyIfNoContainedObjects
CesServiceProfile	AtmSubnetwork	True	True	True	itut_x780::deleteOnlyIfNoContainedObjects
LatestOccurenceLog	AtmSubnetwork	True	True	False	itut_x780::notDeletable
GenericTransportTTP	ManagedElement	True	True	False	itut_x780::notDeletable
ThresholdData	Network	True	True	True	itut_x780::deleteOnlyIfNoContainedObjects
CellProtocolMonCD	AtmLinkEnd	True	True	True	itut_x780::deleteOnlyIfNoContainedObjects
CongDiscardCD	AtmLinkEnd	True	True	True	itut_x780::deleteOnlyIfNoContainedObjects
CongDiscardCD	AtmSubnetwork	True	True	True	itut_x780::deleteOnlyIfNoContainedObjects
PhyPmCDGenericTransportPmCD	AtmLinkEndPhy	True	True	True	itut_x780::deleteOnlyIfNoContainedObjects

[1] In ATM framework manager creates are not allowed even though stated in the namebinding.

PhyPmCDGenericTransportPmCD	GenericTransportTTP	True	True	True	itut_x780::deleteOnlyIfNoContainedObjects
PmOamCD	AtmNetworkCTP	True	True	True	itut_x780::deleteOnlyIfNoContainedObjects
PmOamCD	AtmNetworkTTP	True	True	True	itut_x780::deleteOnlyIfNoContainedObjects
TcAdaptProtMonCD	AtmLinkEnd	True	True	True	itut_x780::deleteOnlyIfNoContainedObjects
TrafficLoadCD	AtmLinkEnd	True	True	True	itut_x780::deleteOnlyIfNoContainedObjects
TrafficLoadCD	AtmSubnetwork	True	True	True	itut_x780::deleteOnlyIfNoContainedObjects
TrafficLoadCD	AtmNetworkCTP	True	True	True	itut_x780::deleteOnlyIfNoContainedObjects
UpcNpcDisagreementsCD	AtmNetworkCTP	True	True	True	itut_x780::deleteOnlyIfNoContainedObjects

3.4. *Interface Creation*

Object Class	manager Creates Allowed	Creation Method(s)
AalProfileTypeOne	True	atmf_m4nw_v2:: AalProfileTypeOneFactory: create; Automatically by Managed System
AalProfileTypeTwo	True	atmf_m4nw_v2:: AalProfileTypeTwoFactory: create; Automatically by Managed System
AalProfileTypeTwoTrunking	True	atmf_m4nw_v2:: AalProfileTypeTwoTrunkingFactory: create; Automatically by Managed System
AalProfileTypeTwoLES	True	atmf_m4nw_v2:: AalProfileTypeTwoLESFactory: create; Automatically by Managed System
AalProfileTypeThreeFour	True	atmf_m4nw_v2:: AalProfileTypeThreeFourFactory: create; Automatically by Managed System
AalProfileTypeFive	True	atmf_m4nw_v2:: AalProfileTypeFiveFactory: create; Automatically by Managed System
AtmImaGroupTP	True	atmf_m4nw_v2:: AtmImaGroupTPFactory: create; Automatically by Managed System
AtmImaLinkTP	True	atmf_m4nw_v2:: AtmImaLinkTPFactory: create; Automatically by Managed System
AtmImaLinkTPPhy	True	atmf_m4nw_v2:: AtmImaLinkTPPhyFactory: create; Automatically by Managed System
AtmLink	True	atmf_m4nw_v2:: AtmLinkFactory:create; Automatically by Managed System

AtmLinkEnd	True	atmf_m4nw_v2:: AtmLinkEndFactory:create; Automatically by Managed System
AtmLinkEndPhy	True	atmf_m4nw_v2:: AtmLinkEndFactory:create; Automatically by Managed System
AtmNetworkAccessProfile	True	atmf_m4nw_v2:: AtmNetworkAccessProfileFactory:create; Automatically by Managed System
AtmNetworkCTP	True	atmf_m4nw_v2:: AtmSubnetwork: setupPtToPtSNCWithLinkEnd; atmf_m4nw_v2:: AtmSubnetwork: setupPtToMultiSNCWithLinkEnd; atmf_m4nw_v2:: AtmSubnetwork: addTpToMultiSNCWithLinkEnd; atmf_m4nw_v2:: AtmNetworkCTPFactory: create;
AtmNetworkTTP	True	atmf_m4nw_v2:: AtmSubnetwork: setupPtToPtSNCWithLinkEnd; atmf_m4nw_v2:: AtmSubnetwork: setupPtToMultiSNCWithLinkEnd; atmf_m4nw_v2:: AtmSubnetwork: addTpToMultiSNCWithLinkEnd; atmf_m4nw_v2:: AtmNetworkTTPFactory: create;
AtmRoutingProfile	True	atmf_m4nw_v2:: AtmRoutingProfileFactory:create;
AtmSNC	True	atmf_m4nw_v2:: AtmSubnetwork: setupPtToPtSNCWithLinkEnd; atmf_m4nw_v2:: AtmSubnetwork: setupPtToPtSNCWithCTP; atmf_m4nw_v2:: AtmSubnetwork: setupPtToMultiSNCWithLinkEnd; atmf_m4nw_v2:: AtmSubnetwork: setupPtToMultiSNCWithCTP; atmf_m4nw_v2:: AtmSubnetwork: addTpToMultiSNCWithLinkEnd; atmf_m4nw_v2:: AtmSubnetwork: addTpToMultiSNCWithCTP;
AtmSubnetwork	False	Automatically by Managed System
AtmTrafficDescABR	True	atmf_m4nw_v2:: AtmTrafficDescABRFactory: create; Automatically by Managed System
AtmTrafficDescCBR	True	atmf_m4nw_v2:: AtmTrafficDescCBRFactory: create; Automatically by Managed System
AtmTrafficDescVBR	True	atmf_m4nw_v2:: AtmTrafficDescVBRFactory: create; Automatically by Managed System
AtmTrafficDescUBR	True	atmf_m4nw_v2:: AtmTrafficDescUBRFactory: create; Automatically by Managed System
AtmTrafficDescGFR	True	atmf_m4nw_v2:: AtmTrafficDescGFRFactory: create; Automatically by Managed System
CesServiceProfile	True	atmf_m4nw_v2:: CesServiceProfileFactory: create; Automatically by Managed

LatestOccurrenceLog	False	
GenericTransportTTP	True	atmf_m4nw_v2:: GenericTransportTTPFactory: create; Automatically by Managed
CellProtocolMonCD	True	atmf_m4nw_v2:: CellProtocolMonCDFactory: create; Automatically by Managed
CongDiscardCD	True	atmf_m4nw_v2:: CongDiscardCDFactory: create; Automatically by Managed
PmOamCD	True	atmf_m4nw_v2:: PmOamCDFactory: create; Automatically by Managed
TcAdaptProtMonCD	True	atmf_m4nw_v2:: TcAdaptProtMonCDFactory: create; Automatically by Managed
TrafficLoadCD	True	atmf_m4nw_v2:: TrafficLoadCDFactory: create; Automatically by Managed
UpcNpcDisagreementsCD	True	atmf_m4nw_v2:: UpcNpcDisagreementsCDFactory: create; Automatically by Managed
GenericTransportPmCD	True	atmf_m4nw_v2:: GenericTransportPmCDFactory: create; Automatically by Managed

4. Summary of Required Object Classes

Table 3-1 summarizes the set of CORBA v2 IDL object interface that is needed to meet the requirements specified in ATM Forum "M4 Interface Requirements and Logical MIB: ATM Network View" version 2 (af-nm-0058.001)[1]. The first column of this table lists the logical managed entities defined [1]. The second column lists attributes and operations associated with the corresponding logical managed entity in column one. The third column lists the IDL object interface that corresponds to the logical managed entity in column one. The fourth column lists the attributes and operations associated with the corresponding IDL object interface in column three. Where appropriate, comments are provided in the fifth column. This table is not intended to exhaustively list every IDL attribute and operation given in Section 5 of this document.

Table 4-1. M4 Network View Logical MIB to CORBA v2 IDL Mapping Table

M4 Logical MIB Managed Entity	**M4 Logical MIB** Attribute/**Operation**	**CORBA v2 IDL Object**	**CORBA IDL** Attribute/**Operation**	**Comment**
Network	Network ID	**M.3120:Network**	**`NameGet`** `(fine grain only)`	
	Relationship with managed resources		*Containment Relationship*	
	query network for contained Managed Entities		*Containment Relationship*	
vcLayerNetworkDomain	atmLNDId	**M.3120:LayerND**	**`NameGet`** `(fine grain only)`	
	Signal Identification		**`signalIdGet`**	
	User Label		**`userLabelGet`** **`userLabelSet`**	
	Relationship with vcTTP		*Containment Relationship via AtmLinkEnd*	
	Relationship with vcTrail		*NO CORBA RELATIONSHIP*	
	Relationship with vcSubnetwork		*Containment Relationship*	
	Relationship with vcTrailRequest		*NO CORBA RELATIONSHIP*	
	query vcLayerNetworkDomain for Delimiting vcTTPs		**NO CORBA OPERATION**	
	query vpLayerNetworkDomain for existing vcTrails		**NO CORBA OPERATION**	
	query vcLayerNetworkDomain for component vcSubnetwork		**NO CORBA OPERATION**	
	setup vcTrail		**NO CORBA OPERATION**	
	setup vcTrailRequest		**NO CORBA OPERATION**	
	addTps to Multipoint Trail		**NO CORBA OPERATION**	
	release vcTrail		**NO CORBA OPERATION**	
	make external vcLinkEnd		**`AtmLinkEndFactory: create`**	
	remove external vcLinkEnd		**AtmLinkEnd: destroy**	
	setup vcTopologicalLink		**AtmLinkFactory: create**	
	release vcTopologicalLink		**AtmLinkEnd:destroy**	
vcLinkConnection	vcLinkConnection ID	**No CORBA Object**		
	Signal Identification			
	Directionality			
	User Label			
	availability Status			
	Administrative State			
	retainedResource			

M4 Logical MIB Managed Entity	**M4 Logical MIB** Attribute/**Operation**	**CORBA v2 IDL Object**	**CORBA IDL** Attribute/**Operation**	**Comment**
	Relationship With vcTopologicalLink		*NO CORBA RELATIONSHIP*	
	Relationship With vcNetworkCTPs		*NO CORBA RELATIONSHIP*	
	query vcLinkConnection for Containing vcTopologicalLink		**NO CORBA OPERATION**	
	query vcLinkConnection for terminating vcNetworkCTPs		**NO CORBA OPERATION**	
vcLinkEnd	vcLinkEnd ID	**AtmLinkEnd**	**NameGet** (fine grain only)	
	Administrative State		**administrativeStateGet administrativeStateSet**	
	Availability Status		**availabilityStatusGet**	
	Egress Maximum Assignable Bandwidth		**egressMaxAssignableBWGet**	
	Ingress Maximum Assignable Bandwidth		**ingressMaxAssignableBWGet**	
	Egress available Bandwidth		**availableLinkEndCapacityGet**	
	Ingress available Bandwidth		**availableLinkEndCapacityGet**	
	User Label		**userLabelGet userLabelSet**	
	Link TP Type		**linkEndTypeGet linkEndTypeSet**	
	NO LOGICAL MIB ATTRIBUTE MATCH		**signalIdGet**	
	Relationship With vcTopologicalLink		**linkPointerGet**	
	Relationship With vcLogicallLinkTP		**logicalLinkEndListGet**	
	Relationship With vcSubnetwork		**AtmSubnetwork: containedLinkEndListGet**	
	Relationship With serverTTPs		**serverTTPListGet**	
	Relationship With vcNetworkAccessProfile		**networkAccessProfileGet networkAccessProfileSet**	
	Relationship With vcNetworkCTP: existing Connection Termination Points		**supportedCTPsGet supportedCTPAdd supportedCTPRemove**	
	query vcLinkEnd for Terminated vcTopologicalLink		**linkPointerGet**	
	query vcLinkEnd for delineated vcSubnetwork		**ContainedInSubnetworkListGet ContainedInSubnetworkListSet ContainedInSubnetworkListAdd containedInSubnetworkListRemove**	
	query vcLinkEnd for associated vpTTP		**serverTTPListGet**	
	associate vcLinkEnd with supporting vpTTP		**serverTTPListGet**	
	vcLinkEnd PVC Trace		**linkPVCTrace**	
vcLogicalLinkTP	vcLogicalLinkTP ID	**AtmLinkEnd**	**NameGet** (fine grain only)	
	Egress Maximum Assignable Bandwidth		**egressMaxAssignableBWGet**	
	Ingress Maximum Assignable Bandwidth		**ingressMaxAssignableBWGet**	
	Egress available Bandwidth		**availableLinkEndCapacityGet**	
	Ingress available Bandwidth		**availableLinkEndCapacityGet**	

M4 Logical MIB Managed Entity	**M4 Logical MIB** Attribute/**Operation**	**CORBA v2 IDL Object**	**CORBA IDL** Attribute/**Operation**	**Comment**
	VCI Range		NetworkAccessProfile: vciRangeGet	
	User Label		**userLabelGet** **userLabelSet**	
	Relationship With vcTopologicalLink		**linkPointerGet**	
	Relationship With vcLinkEnd		**topologicalLinkEndListGet**	
	Relationship With vcSubnetwork		**AtmSubnetwork: containedLinkEndListGet**	
	Relationship With vcNetworkAccessProfile		**networkAccessProfileGet** **networkAccessProfileSet**	
	Relationship With vcNetworkCTP: Existing Connection Termination Points		**supportedCTPsGet** **supportedCTPAdd** **supportedCTPRemove**	
	query vcLogicalLinkTP for terminated vcTopologicalLink		**linkPointerGet**	
	query vcLogicalLinkTP for delineated vcSubnetwork		**ContainedInSubnetworkListGet** **ContainedInSubnetworkListSet** **ContainedInSubnetworkListAdd** **containedInSubnetworkListRemove**	
	query vcLogicalLinkTP for associated vcLinkEnds		**serverTTPListGet**	
	associate vcLogicalLinkTP with supporting vcLinkEnd		**serverTTPListGet**	
	vcLogicalLinkTP PVC Trace		**linkPVCTrace**	
vcNetworkAccessProfile	vcNetworkAccessProfile ID	**AtmNetworkAccessProfile** **AtmNetworkAccessProfileFactory**	**NameGet** (fine grain only)	
	total Egress Bandwidth		**totalEgressBWGet** **totalEgressBWSet**	
	total Ingress Bandwidth		**totalIngressBWGet** **totalIngressBWSet**	
	maximum Number of Active Connection Allowed		**maxNumActiveVcConnGet** **maxNumActiveVcConnSet** **maxNumActiveVpConnGet** **maxNumActiveVpConnSet**	
	VPI/VCI Range		**vpiRangeGet** **vpiRangeSet** **vciRangeGet** **vciRangeSet**	
vcRoutingProfile	vcRoutingProfile ID	**AtmRoutingProfile**	**NameGet** (fine grain only)	
	connectionTypeSupported		**connectionTypeGet**	
	routeDescriptionList		**routeDescriptionListGet**	
	maxHops		**maxHopsGet**	
	Relationship With vcSubnetwork		*Containment Relationship*	
	setup vcRoutingProfile		**AtmRoutingProfileFactory: create**	
vcSubnetwork	Subnetwork ID	**AtmSubnetwork**	**NameGet** (fine grain only)	
	Signal Identification		**signalIdGet**	
	user Label		**userLabelGet** **userLabelSet**	
	availability Status		**availabilityStatusGet**	
	supported by Object List		**supportedByObjectListGet** **supportedByObjectListSet** **supportedByObjectListAdd** **supportedByObjectListRemove**	
	Relationship With vcSubnetworkConnection		Containment Relationship	

M4 Logical MIB Managed Entity	**M4 Logical MIB** Attribute/**Operation**	**CORBA v2 IDL Object**	**CORBA IDL** Attribute/**Operation**	**Comment**
	Relationship With vcSubnetworks		**containedSubnetworkListGet containedSubnetworkListSet containedSubnetworkListAdd containedSubnetworkListRemove**	
	Relationship With vcTopologicalLink		**containedLinkListGet containedLinkListSet containedLinkListAdd containedLinkListRemove**	
	Relationship With vcLogicalLinkTP		**containedLinkEndListGet containedLinkEndListSet containedLinkEndListAdd containedLinkEndListRemove**	
	Relationship With vcLinkEnd		**containedLinkEndListGet containedLinkEndListSet containedLinkEndListAdd containedLinkEndListRemove**	
	query vcSubnetwork for existing vcSubnetworkConnections		**Containment Relationship**	
	query vcSubnetwork for component vcSubnetworks		**containedSubnetworkListGet**	
	query vcSubnetwork for vcTopologicalLinks between its component subnetworks		**containedLinkListGet**	
	query vcSubnetwork for Connecting vcLinkEnds or vcLogicalLinkTPs		**containedLinkEndListGet**	
	setup vcSubnetworkConnection		**setupPtToPtSNCWithCTP setupPtToPtSNCWithLinkEnd setupPtToMultiSNCWithCTP setupPtToMultiSNCWithLinkEnd**	
	modify vcSubnetworkConenction		**AtmNetworkCTP: egressTrafficDescProfileSet; AtmNetworkCTP: ingressTrafficDescProfileSet**	
	addTPs to SunetworkConnection		**addTpToMultiSNCWithCTP addTpToMultiSNCWithLinkEnd removeTpFromMultiSNC**	
	release vcSubnetworkConnection		**releaseSNC**	
vcSubnetworkConnection	vcSubnetworkConnectionID	**AtmSNC**	**NameGet** (fine grain only)	
	Directionality		**directionalityGet**	
	availability Status		**AvailabilityStatusGet operationalStateGet**	
	Administrative Status		**administrativeStateGet administrativeStateSet**	
	User Label		**userLabelGet userLabelSet**	
	restorableIndicator		**protectedGet**	
	retainedResource		**NO CORBA OPERATION**	
	provisionType		**provisionTypeGet provisionTypeSet**	
	NO LOGICAL MIB ATTRIBUTE MATCH		**signalIdGet connectionTypeGet**	
	Relationship With networkCTPs		**aEndNetworkTPListGet zEndNetworkTPListGet**	
	Relationship With subnetworkConnections		**componentPointerListGet**	

M4 Logical MIB Managed Entity	**M4 Logical MIB** Attribute/**Operation**	**CORBA v2 IDL Object**	**CORBA IDL** Attribute/**Operation**	**Comment**
	Relationship With vcLinkConnections		**NO CORBA OPERATION**	
	Relationship With routingProfiles		**relatedRoutingProfileGet**	
	query subnetworkConnection for terminating networkCTPs		**aEndNetworkTPListGet zEndNetworkTPListGet**	
	query vcSubnetworkConnection for Component vcSubnetworkConnections		**componentPointerListGet**	
	vcSubnetworkConnection Connection Trace		**traceSNC**	
vcTopologicalLink	vcTopologicalLink ID	**AtmLink**	**NameGet** (fine grain only)	
	Signal Identification		**signalIdGet**	
	Directionality		**NO CORBA OPERATION**	
	Operational State		**AvailabilityStatusGet**	
	provisioned Bandwidth		**AtmNetworkAccessProfile: totalIngressBWGet**	
	available Bandwidth		**availableLinkCapacityGet**	
	restorationMode		**restorationModeGet restorationModeSet**	
	Customer Identification		**userLabelGet userLabelSet**	
	Weight		**usageCostGet usageCostSet**	
	NO LOGICAL MIB ATTRIBUTE MATCH		**administrativeStateGet administrativeStateSet**	
	Relationship With linkConnections		**NO CORBA OPERATION**	
	Relationship With logicalLinkTP		**aEndGet zEndGet**	
	Relationship With linkEnd		**aEndGet zEndGet**	
	Relationship With subnetwork		NO CORBA OPERATION (via Link Ends) *** replaced operation	
	Relationship With vcNetworkAccessProfile		**NO CORBA OPERATION (via Link Ends) *** replaced operation**	
	query vcTopologicalLink for Contained vcLinkConnections		**NO CORBA OPERATION**	
	query vcTopologicalLink For Terminating vcLinkEnds or vcLogicalLinkTPs		**aEndGet zEndGet**	
	query vcTopologicalLink For Delineated vcSubnetworks		NO CORBA OPERATION (via Link Ends) *** replaced operation	
	set up vcLinkConnection		**NO CORBA OPERATION**	
	modify vcLinkConnection		**NO CORBA OPERATION**	
	release vcLinkConnection		**NO CORBA OPERATION**	
	vcTopologicalLink PVC Trace		**linkPVCTrace**	
vcTrail	vcTrail ID	**NO M4 IDL OBJECT (Trails are represented as AtmSNCs with associated TTPs)**		
	signal Identification			
	Directionality			
	User Label			
	Administrative State			
	availability State			
	restorable Indicator			
	retainedResource			
	Relationship With vcNetworkTTP			

M4 Logical MIB Managed Entity	M4 Logical MIB Attribute/**Operation**	CORBA v2 IDL Object	CORBA IDL Attribute/**Operation**	Comment
	query vcTrail for terminatingTTPs			
	vcTrail Connection Trace			
vcTrailRequest	vpTrailrequest ID	**NO M4 IDL OBJECT**		
	Request Status			
	requestType			
	requestCommittedTime			
	Relationship With vcTrail			
vpLayerNetworkDomain	Signal Identification	**AtmLND**	**`signalIdGet`**	
	NO LOGICAL MIB ATTRIBUTE MATCH			
	User Label		**`userLabelGet`** **`userLabelSet`**	
	NO LOGICAL MIB ATTRIBUTE MATCH			
	Relationship With vpTTP		*Containment Relationship via AtmLinkEnd*	
	Relationship With vpTrail		*NO CORBA RELATIONSHIP*	
	Relationship With vpSubnetwork		*Containment Relationship*	
	Relationship With vpTrailRequest		*NO CORBA RELATIONSHIP*	
	query vpLayerNetworkDomain for Delimiting vpTTPs		**NO CORBA OPERATION**	
	query vpLayerNetworkDomain for existing vpTrails		**NO CORBA OPERATION**	
	query vpLayerNetworkDomain for component vpSubnetwork		*Containment Relationship*	
	setup vpTrail		**NO CORBA OPERATION**	
	setup vpTrailRequest		**NO CORBA OPERATION**	
	addTps To Multipoint Trail		**NO CORBA OPERATION**	
	release vpTrail		**NO CORBA OPERATION**	
	Make External vpLinkEnd		**`AtmLinkEndFactory: create`**	
	Remove External vpLinkEnd		**AtmLinkEnd:destroy**	
	setup vpTopologicalLink		**AtmLinkFactory: create**	
	release vpTopologicalLink		**AtmLinkEnd:destroy**	
vpLinkConnection	vpLinkConnection ID	**No CORBA Object**		
	Signal Identification			
	Directionality			
	User Label			
	availability Status			
	Administrative State			
	retainedResource			
	Relationship With vpTopologicalLink			
	Relationship With vpNetworkCTPs			
	query vpLinkConnection for containing vpTopologicalLink			
	query vpLinkConnection For terminating vpNetworkCTPs			
vpLinkEnd	vpLinkEnd ID	**AtmLinkEnd**	**`NameGet`** `(fine grain only)`	
	Administrative State		**`administrativeStateGet`** **`administrativeStateSet`**	
	Availability Status		**`availabilityStatusGet`**	
	Egress Maximum Assignable Bandwidth		**`egressMaxAssignableBWGet`**	

M4 Logical MIB Managed Entity	**M4 Logical MIB** Attribute/**Operation**	**CORBA v2 IDL Object**	**CORBA IDL** Attribute/**Operation**	**Comment**
	Ingress Maximum Assignable Bandwidth		**ingressMaxAssignableBWGet**	
	Egress available Bandwidth		**availableLinkEndCapacityGet**	
	Ingress available Bandwidth		**availableLinkEndCapacityGet**	
	User Label		**userLabelGet** **userLabelSet**	
	Link TP Type		**linkEndTypeGet** **linkEndTypeSet**	
	Loopback Location Identifier		**loopbackLocIDGet** **loopbackLocIDSet**	
	ILMI Virtual Identifier		**ilmiVpiVciGet** **ilmiVpiVciSet**	
	Supporting NE Location		**AtmLinkEndPhy:** **serverTTPPortIDGet**	
	Supporting Circuit Pack Location		**AtmLinkEndPhy:** **serverTTPPortIDGet**	
	Server TTP Name		**serverTTPListGet**	
	Server TTP Characteristic Information Type		**AtmLinkEndPhy:** **serverTTPCharInfoGet**	
	Server TTP Port Id		**AtmLinkEndPhy:** **serverTTPPortIDGet,** **serverTTPPortIDSet**	
	Server TTP Operational State		**AtmLinkEndPhy:** **serverTTPOpStateGet**	
	Server TTP Technology Specific Additional Information		**vendorProfileListGet** **vendorProfileListAdd** **vendorProfileListRemove**	
	Cell Scrambling Enable		**cellScramblingEnabledGet** **cellScramblingEnabledSet**	
	Subscriber Address		**NO CORBA SUPPORT**	
	Prefered Carrier		**NO CORBA SUPPORT**	
	NO LOGICAL MIB ATTRIBUTE MATCH		**signalIdGet** **ilmiEstabConnectivityPollIntervalGet** **ilmiEstabConnectivityPollIntervalSet** **ilmiCheckConnectivityPollIntervalGet** **ilmiCheckConnectivityPollIntervalSet** **ilmiConnectivityPollFactorGet** **ilmiConnectivityPollFactorSet** **AtmLinkEndPhy:** **serverTTPCharInfoGet** **serverTTPPortIDSet** **serverTTPOpStateGet** **profileGet** **profileSet** **currentProblemListGet**	
	Relationship With vpTopologicalLink		**linkPointerGet**	
	Relationship With vpLogicalLinkTP		**logicalLinkEndListGet**	
	Relationship With vpSubnetwork		**AtmSubnetwork:** **containedLinkEndListGet**	
	Relationship With serverTTPs		**serverTTPListGet**	
	Relationship With vpNetworkAccessProfile		**networkAccessProfileGet** **networkAccessProfileSet**	
	Relationship With vpNetworkCTP: Existing Connection Termination Points		**supportedCTPsGet** **supportedCTPAdd** **supportedCTPRemove**	

M4 Logical MIB Managed Entity	M4 Logical MIB Attribute/**Operation**	CORBA v2 IDL Object	CORBA IDL Attribute/**Operation**	Comment
	query vpLinkEnd forTerminated vpTopologicalLink		**linkPointerGet**	
	query vpLinkEnd for delineated vpSubnetwork		**ContainedInSubnetworkList Get** **ContainedInSubnetworkList Set** **ContainedInSubnetworkList Add** **containedInSubnetworkList Remove**	
	query vpLinkEnd For associated serverTTP		**serverTTPListGet**	
	associate vpLinkEnd with supporting serverTTP		**serverTTPListGet**	
	vpLinkEnd PVC Trace		**linkPVCTrace**	
vpLogicalLinkTP	vpLogicalLinkTP ID	**AtmLinkEnd**	**NameGet** (fine grain only)	
	Egress Maximum Assignable Bandwidth		**egressMaxAssignableBWGet**	
	Ingress Maximum Assignable Bandwidth		**ingressMaxAssignableBWGet**	
	Egress available Bandwidth		**availableLinkEndCapacityGet**	
	Ingress available Bandwidth		**availableLinkEndCapacityGet**	
	VPI Range		NetworkAccessProfile: vciRangeGet	
	User Label		**userLabelGet** **userLabelSet**	
	Relationship With vpTopologicalLink		**linkPointerGet**	
	Relationship With vpLinkEnd		**topologicalLinkEndListGet**	
	Relationship With vpSubnetwork		**AtmSubnetwork: containedLinkEndListGet**	
	Relationship With vpNetworkAccessProfile		**networkAccessProfileGet** **networkAccessProfileSet**	
	Relationship With vpNetworkCTP: Existing Connection Termination Points		**supportedCTPsGet** **supportedCTPAdd** **supportedCTPRemove**	
	query vpLogicalLinkTP for terminated vpTopologicalLink		**linkPointerGet**	
	query vpLogicalLinkTP for delineated vpSubnetwork		**ContainedInSubnetworkList Get** **ContainedInSubnetworkList Set** **ContainedInSubnetworkList Add** **containedInSubnetworkList Remove**	
	query vpLogicalLinkTP for associated vpLinkEnds		**serverTTPListGet**	
	associate vpLogicalLinkTP with supporting vpLinkEnd		**serverTTPListGet**	
	vpLogicalLinkTP PVC Trace		**linkPVCTrace**	
vpNetworkAccessProfile	vpNetworkAccessProfile ID	**AtmNetworkAccessProfile** **AtmNetworkAccessProfileFactory**	**NameGet** (fine grain only)	
	total Egress Bandwidth		**totalEgressBWGet** **totalEgressBWSet**	
	total Ingress Bandwidth		**totalIngressBWGet** **totalIngressBWSet**	
	maximum Number of Active Connection Allowed		**maxNumActiveVcConnGet** **maxNumActiveVcConnSet** **maxNumActiveVpConnGet** **maxNumActiveVpConnSet**	

M4 Logical MIB Managed Entity	M4 Logical MIB Attribute/**Operation**	CORBA v2 IDL Object	CORBA IDL Attribute/**Operation**	Comment
	VPI/VCI Range		**vpiRangeGet** **vpiRangeSet** **vciRangeGet** **vciRangeSet**	
vpRoutingProfile	vpRoutingProfile ID	**AtmRoutingProfile**	**NameGet** (fine grain only)	
	connectionTypeSupported		**connectionTypeGet**	
	routeDescriptionList		**routeDescriptionListGet**	
	maxHops		**maxHopsGet**	
	Relationship With vpSubnetwork		*Containment Relationship*	
	setup vpRoutingProfile		**AtmRoutingProfileFactory: create**	
vpSubnetwork	Subnetwork ID	**AtmSubnetwork**	**NameGet** (fine grain only)	
	Signal Identification		**signalIdGet**	
	user Label		**userLabelGet** **userLabelSet**	
	availability Status		**availabilityStatusGet**	
	supported by Object List		**supportedByObjectListGet** **supportedByObjectListSet** **supportedByObjectListAdd** **supportedByObjectListRemove**	
	Relationship with vpSubnetworkConnection		Containment Relationship	
	Relationship with vpSubnetworks		**containedSubnetworkListGet** **containedSubnetworkListSet** **containedSubnetworkListAdd** **containedSubnetworkListRemove**	
	Relationship with vpTopologicalLink		**containedLinkListGet** **containedLinkListSet** **containedLinkListAdd** **containedLinkListRemove**	
	Relationship with vpLogicalLinkTP		**containedLinkEndListGet** **containedLinkEndListSet** **containedLinkEndListAdd** **containedLinkEndListRemove**	
	Relationship with vpLinkEnd		**containedLinkEndListGet** **containedLinkEndListSet** **containedLinkEndListAdd** **containedLinkEndListRemove**	
	query vpSubnetwork for delimiting vpNetworkCTPs		AtmLinkEnd: supportedCTPsGet	
	query vpSubnetwork for existing vpSubnetworkConnections		Containment Relationship	
	query vpSubnetwork for component vpSubnetworks		**containedSubnetworkListGet**	
	query vpSubnetwork for vpTopologicalLinks between its component subnetworks		**containedLinkListGet**	
	query vpSubnetwork For Connecting vpLinkEnds or vpLogicalLinkTPs		**containedLinkEndListGet**	
	setup vpSubnetworkConnection		**setupPtToPtSNCWithCTP** **setupPtToPtSNCWithLinkEnd** **setupPtToMultiSNCWithCTP** **setupPtToMultiSNCWithLinkEnd**	

M4 Logical MIB Managed Entity	M4 Logical MIB Attribute/**Operation**	CORBA v2 IDL Object	CORBA IDL Attribute/**Operation**	Comment
	modify vpSubnetworkConenction		**AtmNetworkCTP: egressTrafficDescProfileSet; AtmNetworkCTP: ingressTrafficDescProfileSet**	
	addTPs to SunetworkConnection		**addTpToMultiSNCWithCTP addTpToMultiSNCWithLinkEnd removeTpFromMultiSNC**	
	release vpSubnetworkConnection		**releaseSNC**	
vpSubnetworkConnection	Directionality	**AtmSNC**	**directionalityGet**	
	availability Status		**AvailabilityStatusGet operationalStateGet**	
	Administrative Status		**administrativeStateGet administrativeStateSet**	
	User Label		**userLabelGet userLabelSet**	
	restorableIndicator		**protectedGet**	
	retainedResource		**NO CORBA OPERATION**	
	provisionType		**provisionTypeGet provisionTypeSet**	
	NO LOGICAL MIB ATTRIBUTE MATCH		**signalIdGet connectionTypeGet**	
	Relationship with networkCTPs		**aEndNetworkTPListGet zEndNetworkTPListGet**	
	Relationship with subnetworkConnections		**componentPointerListGet**	
	Relationship with vpLinkConnections		**NO CORBA OPERATION**	
	Relationship with routingProfiles		**relatedRoutingProfileGet**	
	query subnetworkConnection for terminating networkCTPs		**aEndNetworkTPListGet zEndNetworkTPListGet**	
	query vpSubnetworkConnection for Component vpSubnetworkConnections		**componentPointerListGet**	
	vpSubnetworkConnection Connection Trace		**traceSNC**	
vpTopologicalLink	vpTopologicalLink ID	**AtmLink**	**NameGet** (fine grain only)	
	Signal Identification		**signalIdGet**	
	Directionality		**NO CORBA OPERATION**	
	Operational State		**AvailabilityStatusGet**	
	provisioned Bandwidth		**AtmNetworkAccessProfile: totalIngressBWGet; AtmNetworkAccessProfile: totalEgressBWGet;**	
	available Bandwidth		**availableLinkCapacityGet**	
	restorationMode		**restorationModeGet restorationModeSet**	
	Customer Identification		**userLabelGet userLabelSet**	
	Weight		**usageCostGet usageCostSet**	
	NO LOGICAL MIB ATTRIBUTE MATCH		**administrativeStateGet administrativeStateSet**	
	Relationship With linkConnections		**NO CORBA OPERATION**	
	Relationship with logicalLinkTP		**aEndGet zEndGet**	
	Relationship with linkEnd		**aEndGet zEndGet**	
	Relationship with subnetwork		NO CORBA OPERATION (via Link Ends) *** replaced operation	

M4 Logical MIB Managed Entity	M4 Logical MIB Attribute/**Operation**	CORBA v2 IDL Object	CORBA IDL Attribute/**Operation**	Comment
	Relationship with vpNetworkAccessProfile		**NO CORBA OPERATION (via Link Ends) *** replaced operation**	
	query vpTopologicalLink for contained vpLinkConnections		**aEndGet zEndGet**	
	query vpTopologicalLink for Terminating vpLinkEnds or vpLogicalLinkTPs		NO CORBA OPERATION (via Link Ends) *** replaced operation	
	query vpTopologicalLink for delineated vpSubnetworks			
	set up vpLinkConnection		**NO CORBA OPERATION**	
	modify vpLinkConnection		**NO CORBA OPERATION**	
	release vpLinkConnection		**NO CORBA OPERATION**	
	vpTopologicalLink PVC Trace		**linkPVCTrace**	
vpTrail	vpTrail ID	**NO M4 IDL OBJECT (Trails are represented as AtmSNCs with associated TTPs)**		
	signal Identification			
	Directionality			
	User Label			
	Administrative State			
	availability State			
	restorable Indicator			
	retained Resource			
	Relationship with vpNetworkTTP			
	query vpTrail for terminatingTTPs			
	vpTrail Connection Trace			
	vpTrailrequest ID			
vpTrailRequest	Request Status	**NO M4 IDL OBJECT**		
	requestType			
	requestCommittedTime			
	Relationship With vpTrail			
vcNetworkCTP	vcCTP ID	**AtmNetworkCTP**	**NameGet** (fine grain only)	
	VPI/VCI Value		**networkTPVpiVciGet**	
	User Label		**NO CORBA OPERATION**	
	segment endpoint		**segmentEndpointGet segmentEndpointSet**	
	Ingress Tagging Indicator		**ingressTaggingIndGet ingressTaggingIndSet**	
	Egress Tagging Indicator		**egressTaggingIndGet egressTaggingIndSet**	
	PM OAM Method		**pmOamMethodGet pmOamMethodSet**	
	PM OAM Direction		**pmOamDirectionGet pmOamDirectionSet**	
	PM OAM block size		**pmOamBlockSizeGet pmOamBlockSizeSet**	
	PM OAM Forward Active		**pmOamForwardActiveGet pmOamForwardActiveSet**	
	PM OAM Backward Active		**pmOamBackwardActiveGet pmOamBackwardActiveSet**	
	NO LOGICAL MIB ATTRIBUTE MATCH		**signalIdGet operationalStateGet alarmStatusGet currentProblemListGet alarmSeverityAssignmentProfilePointerGet alarmSeverityAssignmentProfilePointerSet pointDirectionalityGet segmentEndpointGet segmentEndpointSet**	

M4 Logical MIB Managed Entity	**M4 Logical MIB** Attribute/**Operation**	**CORBA v2 IDL Object**	**CORBA IDL** Attribute/**Operation**	**Comment**
	Relationship with vcNetworkTTP		**relatedAtmTTPGet** **relatedAtmTTPSet**	
	Relationship with subnetworkConnection		**sncPointerGet**	
	Relationship with trafficDescriptorProfile		**egressTrafficDescProfileGet** **egressTrafficDescProfileSet** **ingressTrafficDescProfileGet** **ingressTrafficDescProfileSet**	
	associate vcNetworkCTP with vcNetworkTTP		**relatedAtmTTPSet**	
	query vcNetworkCTP for associated vcNetworkTTP		**relatedAtmTTPGet**	
	query vcNetworkCTP for associated subnetworkConnections		**sncPointerGet**	
	lookback vcTrail at vcNetworkCTP		**loopbackOamCell**	
vcNetworkTTP	vcTTP ID	**AtmNetworkTTP**	**NameGet** (fine grain only)	
	availability Status		**operationalStateGet**	
	PM OAM Method		**pmOamMethodGet** **pmOamMethodSet**	
	PM OAM Direction		**pmOamDirectionGet** **pmOamDirectionSet**	
	PM OAM block size		**pmOamBlockSizeGet** **pmOamBlockSizeSet**	
	PM OAM Forward Active		**pmOamForwardActiveGet** **pmOamForwardActiveSet**	
	PM OAM Backward Active		**getPmOamBackwardActive** **setPmOamBackwardActive**	
	NO LOGICAL MIB ATTRIBUTE MATCH		**networkTPVpiVciGet** **signalIdGet** **operationalStateGet** **alarmStatusGet** **currentProblemListGet** **alarmSeverityAssignmentProfilePointerGet** **alarmSeverityAssignmentProfilePointerSet** **pointDirectionalityGet** **segmentEndpointGet** **segmentEndpointSet** **clientLinkEndPointerListGet** **clientLinkEndPointerListSet**	
	Relationship with vcNetworkCTP		**relatedAtmCTPGet** **relatedAtmCTPSet**	
	Relationship with vcTrail		**sncPointerGet**	
	Relationship with AAL Profile		**aalProfilePointerGet** **aalProfilePointerSet**	
	Relationship with Service Profile		**serviceProfilePointerGet** **serviceProfilePointerSet**	
	query vcNetworkTTP for associated vcNetworkCTP		**relatedAtmCTPGet**	
	query vcTTP For terminated vcTrail		**sncPointerGet**	
	loopback vpTrail at vpTTP		**loopbackOamCell**	
vpNetworkCTP	vpCTP ID	**AtmNetworkCTP**	**NameGet** (fine grain only)	
	VPI Value		**networkTPVpiVciGet**	
	User Label		**NO CORBA OPERATION**	
	segment endpoint		**segmentEndpointGet** **segmentEndpointSet**	

M4 Logical MIB Managed Entity	**M4 Logical MIB** Attribute/**Operation**	**CORBA v2 IDL Object**	**CORBA IDL** Attribute/**Operation**	**Comment**
	Ingress Tagging Indicator		**ingressTaggingIndGet** **ingressTaggingIndSet**	
	Egress Tagging Indicator		**egressTaggingIndGet** **egressTaggingIndSet**	
	PM OAM Method		**pmOamMethodGet** **pmOamMethodSet**	
	PM OAM Direction		**pmOamDirectionGet** **pmOamDirectionSet**	
	PM OAM block size		**pmOamBlockSizeGet** **pmOamBlockSizeSet**	
	PM OAM Forward Active		**pmOamForwardActiveGet** **pmOamForwardActiveSet**	
	PM OAM Backward Active		**pmOamBackwardActiveGet** **pmOamBackwardActiveSet**	
	NO LOGICAL MIB ATTRIBUTE MATCH		**signalIdGet** **operationalStateGet** **alarmStatusGet** **currentProblemListGet** **alarmSeverityAssignmentProfilePointerGet** **alarmSeverityAssignmentProfilePointerSet** **pointDirectionalityGet** **segmentEndpointGet** **segmentEndpointSet**	
	Relationship with vpNetworkTTP		**relatedAtmTTPGet** **relatedAtmTTPSet**	
	Relationship with subnetworkConnection		**sncPointerGet**	
	Relationship with trafficDescriptorProfile		**egressTrafficDescProfileGet** **egressTrafficDescProfileSet** **ingressTrafficDescProfileGet** **ingressTrafficDescProfileSet**	
	associate vpNetworkCTP with vpNetworkTTP		**relatedAtmTTPSet**	
	query vpNetworkCTP for associated vpNetworkTTP		**relatedAtmTTPGet**	
	query vpNetworkCTP for associated subnetworkConnections		**sncPointerGet**	
	lookback vpTrail at vpNetworkCTP		**loopbackOamCell**	
vpNetworkTTP	vpTTP ID	**AtmNetworkTTP**	**NameGet** (fine grain only)	
	availability Status		**operationalStateGet**	
	PM OAM Method		**pmOamMethodGet** **pmOamMethodSet**	
	PM OAM Direction		**pmOamDirectionGet** **pmOamDirectionSet**	
	PM OAM block size		**pmOamBlockSizeGet** **pmOamBlockSizeSet**	
	PM OAM Forward Active		**pmOamForwardActiveGet** **pmOamForwardActiveSet**	
	PM OAM Backward Active		**getPmOamBackwardActive** **setPmOamBackwardActive**	

M4 Logical MIB Managed Entity	**M4 Logical MIB** Attribute/**Operation**	**CORBA v2 IDL Object**	**CORBA IDL** Attribute/**Operation**	**Comment**
	NO LOGICAL MIB ATTRIBUTE MATCH		**networkTPVpiVciGet** **signalIdGet** **operationalStateGet** **alarmStatusGet** **currentProblemListGet** **alarmSeverityAssignmentProfilePointerGet** **alarmSeverityAssignmentProfilePointerSet** **pointDirectionalityGet** **segmentEndpointGet** **segmentEndpointSet** **clientLinkEndPointerListGet** **clientLinkEndPointerListSet**	
	Relationship with vpNetworkCTP		**relatedAtmCTPGet** **relatedAtmCTPSet**	
	Relationship with vpTrail		**sncPointerGet**	
	query vpNetworkTTP for associated vpNetworkCTP		**relatedAtmCTPGet**	
	query vpTTP For terminated vpTrail		**sncPointerGet**	
	loopback vpTrail at vpTTP		**loopbackOamCell**	
aal1Profile		**AalProfileTypeOne**	**aalTypeGet** **cbrRateGet** **cellLossIntegrationPeriodGet** **clockRecoveryTypeGet** **forwardErrorCorrectionMethodGet** **partiallyFilledCellsGet** **structuredDataTransferGet** **subTypeGet**	

M4 Logical MIB Managed Entity	**M4 Logical MIB** Attribute/**Operation**	**CORBA v2 IDL Object**	**CORBA IDL** Attribute/**Operation**	**Comment**
aal2Profile		**AalProfileTypeTwo**	**aalTypeGet** **applicationIdentifierGet** **configResponsibilityGet** **cpsMaxMultiplexedChannelsGet** **cpsMaxSduLengthGet** **sscsMaxSssarSduLengthGet** **sscsSstedStatusGet** **sscsSsadtStatusGet** **sscsServiceCategoryGet** **sscsAudioServiceTransportGet** **sscsProfileSourceGet** **sscsIeeeOuiGet** **sscsPredefinedProfileIdentifierGet** **sscsPcmEncodingGet** **sscsFaxDemodulationTransportGet** **sscsCasSignalingTransportGet** **sscsDtmfDigitPacketTransportGet** **sscsMfR1DigitPacketTransportGet** **sscsMfR2DigitPacketTransportGet** **sscsCircuitModeDataTransportGet** **sscsCircuitModeDataNumChannelsGet** **sscsFrameModeDataTransportGet** **sscsFrameModeDataMaxLengthGet** **SscopSduLengthGet** **sscopUuLengthGet** **(aal2 Trunking adds)** **vcciGet** **signalingVcciGet** **(aal2 LES adds)** **cpsCIDLowerLimitGet** **cpsCIDUpperLimitGet** **cpsOptimisationGet**	
aal3/4Profile		**AalProfileTypeThreeFour**	**aalTypeGet** **maxCpcsSduSizeForwardGet** **maxCpcsSduSizeBackwardGet** **midRangeLowGet** **midRangeHighGet** **aalModeGet** **sscsTypeGet**	
aal5Profile		**AalProfileTypeFive**	**aalTypeGet** **maxCpcsSduSizeForwardGet** **maxCpcsSduSizeBackwardGet** **aalModeGet** **sscsTypeGet**	
alarmRecord		**OMG Telecom Log Service: Log**		
alarmSeverityAssignmentProfile		**M.3120: AlarmSeverityAssignmentProfile**	**alarmSeverityAssignmentListGet** **alarmSeverityAssignmentListSet** **alarmSeverityAssignmentListAdd** **alarmSeverityAssignmentListRemove**	
atmCellProtocolMonitoringLogRecord		**NO M4 IDL OBJECT**		

M4 Logical MIB Managed Entity	M4 Logical MIB Attribute/**Operation**	CORBA v2 IDL Object	CORBA IDL Attribute/**Operation**	Comment
cesServiceProfile		**CesServiceProfile**	**cesBufferedCDVToleranceGet** **cesBufferedCDVToleranceSet** **channelAssociatedSignallingGet** **channelAssociatedSignallingSet**	
eventForwardingDiscriminator		**CosNotifyFilter::Filter**		
latestOccurrenceLog		**NO M4 IDL OBJECT**		
log		**OMG Telecom Log Service: Log**		See Appendix C, Table C-1
trafficDescriptor	Managed Entity ID	**AtmAbstractTrafficDesc**	**NameGet (fine grain only)**	AtmTrafficDesc is uninstantiable. See Note 2 for Table 3-1.
	Profile Name		**profileNameGet**	
	Service Category		**serviceCategoryGet**	
	Conformance Definition		**conformanceDefinitionGet**	
	Peak Cell Rate - Ingress and Egress		**peakCellRateGet**	See Note 2
	Cell Delay Variation Tolerance in relation to the PCR - Ingress and Egress		**cDVTolerancePCRGet**	
	Sustainable Cell Rate - Ingress and Egress	**AtmTrafficDescVBR**	**vBRSustainableCellRate**	
	Maximum Burst Size - Ingress and Egress		**vBRMaxBurstSize**	
	Minimum Cell Rate - Ingress and Egress	**AtmTrafficDescABR**	**minCellRateGet**	
	Initial Cell Rate - Ingress and Egress		**initialCellRateGet**	
	Transient Buffer Exposure - Ingress and Egress		**transientBufferExposureGet**	
	Rate Decrease Factor - Ingress and Egress		**rateDecreaseFactorGet**	
	Rate Increase Factor - Ingress and Egress		**rateIncreaseFactorGet**	
	Fixed Round Trip Time		**fixedRoundTripTimeGet**	
	Nrm - Ingress and Egress		**aBRNrmGet**	
	Trm - Ingress and Egress		**aBRTrmGet**	
	CDF - Ingress and Egress		**aBRCDFGet**	
	ADTF - Ingress and Egress		**aBRADTFGet**	
	CLR - Ingress and Egress	**AtmTrafficDescCBR**	**cBRCLRGet**	
		AtmTrafficDescVBR	**vBRCLRGet**	
Performance Management Objects				
atmCellProtocolMonitoringCurrentData	Managed Entity ID	**CellProtocolMonCurrentData**	**NameGet** (fine grain only)	**Attributes from CurrentData in Q.822.1**
	Administrative State		**administrativeStateGet** **administrativeStateSet**	
	Suspect Flag		**suspectIntervalFlagGet**	
	Elapsed Time		**elapsedTimeGet**	
	Threshold Data ID		**thresholdDataInstanceListGet** **thresholdDataInstanceListSet**	
	Number of Suppressed Intervals		**numSuppressedIntervalsGet**	
	No M4 attribute; from Q.822.		**operationalStateGet**	
	No M4 attribute; from Q.822.		**granularityPeriodGet**	
	Discarded Cells due to Protocol Errors		NumberDiscCellsProtErr	
	Received OAM Cells		Number RecvOAMCells	
	No actions have been defined.		**historyRetentionSet** **historyRetentionGet**	

M4 Logical MIB Managed Entity	**M4 Logical MIB** Attribute/**Operation**	**CORBA v2 IDL Object**	**CORBA IDL** Attribute/**Operation**	**Comment**
atmCellProtocolMonitoringHistoryData	Managed Entity ID	**CellProtocolMonCurrentData**	**NameGet** (fine grain only)	Attributes from HistoryDataValueType ofCurrentData in Q.822.1
	Period End Time		**HistoryDataValueType: periodEndTime**	
	Suspect Flag		**HistoryDataValueType: suspectIntervalFlag**	
	Number of Suppressed Intervals		**HistoryDataValueType: numSupressedIntevals**	
	No M4 attribute; from Q.822.		**HistoryDataValueType: granularityPeriod**	
	Discarded Cells due to Protocol Errors		NumberDiscCellsProtErr	
	Received OAM Cells		NumberRecvOAMCells	
	No actions have been defined.		**getMostRecent**	
			getBetween	
atmTrafficLoadCurrentData	Managed Entity ID	**AtmTrafficLoadCurrentData**	**NameGet** (fine grain only)	Attributes from CurrentData in Q.822.1
	Administrative State		**administrativeStateGet administrativeStateSet**	
	Suspect Flag		**suspectIntervalFlagGet**	
	Elapsed Time		**elapsedTimeGet**	
	Threshold Data ID		**thresholdDataInstanceListGet thresholdDataInstanceListSet**	
	Number of Suppressed Intervals		**numSuppressedIntervalsGet**	
	No M4 attribute; from Q.822.		**operationalStateGet**	
	No M4 attribute; from Q.822.		**granularityPeriodGet**	
	Cells Received		NumberCellsRecvd	
	Cells Transmitted		NumberCellsTrnsd	
	No actions have been defined.		**historyRetentionSet historyRetentionGet**	
atmTrafficLoadHistoryData	Managed Entity ID	**AtmTrafficLoadCurrentData**	**NameGet** (fine grain only)	Attributes from HistoryDataValueType ofCurrentData in Q.822.1
	Period End Time		**HistoryDataValueType: periodEndTime**	
	Suspect Flag		**HistoryDataValueType: suspectIntervalFlag**	
	Number of Suppressed Intervals		**HistoryDataValueType: numSupressedIntevals**	
	No M4 attribute; from Q.822.		**HistoryDataValueType: granularityPeriod**	
	Cells Received		NumberCellsRecvd	
	Cells Transmitted		NumberCellsTrnsd	
	No actions have been defined.		**getMostRecent**	
			getBetween	
congestionDiscardCurrentData	Managed Entity ID	**CongDiscardCurrentData**	**NameGet** (fine grain only)	Attributes from CurrentData in Q.822.1
	Administrative State		**administrativeStateGet administrativeStateSet**	
	Suspect Flag		**suspectIntervalFlagGet**	
	Elapsed Time		**elapsedTimeGet**	
	Threshold Data ID		**thresholdDataInstanceListGet thresholdDataInstanceListSet**	
	Number of Supprressed Intervals		**numSuppressedIntervalsGet**	
	No M4 attribute; from Q.822.		**operationalStateGet**	
	No M4 attribute; from Q.822.		**granularityPeriodGet**	
	All Cells Discarded		TcountAllCellsDisc	
	Priority Cells Discarded		TcountPriorityCellsDisc	
	No actions have been defined.		**historyRetentionSet historyRetentionGet**	
	Managed Entity ID		**NameGet** (fine grain only)	

M4 Logical MIB Managed Entity	M4 Logical MIB Attribute/**Operation**	CORBA v2 IDL Object	CORBA IDL Attribute/**Operation**	Comment
	Managed Entity ID		**NameGet** (fine grain only)	
congestionDiscardHistoryData	Suspect Flag Period End Time	**CongDiscardCurrentData**	**HistoryDataValueType: suspectIntervalFlag** **HistoryDataValueType: periodEndTime**	Attributes from HistoryDataValueType ofCurrentData in Q.822.1
	Number of Suppressed Intervals		**HistoryDataValueType: numSupressedIntevals**	
	No M4 attribute; from Q.822.		**HistoryDataValueType: periodEndTime**	
	All Cells Discarded		TcountAllCellsDisc	
	Priority Cells Discarded		TcountPriorityCellsDisc	
	No actions have been defined.		**getMostRecent** **getBetween**	
tcAdaptorProtocolMonitoringCurrentData	Managed Entity ID	**TcAdaptProtMonCurrentData**	**NameGet** (fine grain only)	Attributes from CurrentData in Q.822.1
	Administrative State		**administrativeStateGet** **administrativeStateSet**	
	Suspect Flag		**suspectIntervalFlagGet**	
	Elapsed Time		**elapsedTimeGet**	
	Threshold Data ID		**thresholdDataInstanceListGet** **thresholdDataInstanceListSet**	
	Number of Suppressed Intervals		**numSuppressedIntervalsGet**	
	No M4 attribute; from Q.822.		**operationalStateGet**	
	No M4 attribute; from Q.822.		**granularityPeriodGet**	
	Discarded Cells due to HEC violation		NumberDiscCellsHECViolat	
	No actions have been defined.		**historyRetentionSet** **historyRetentionGet**	
tcAdaptorProtocolMonitoringHistoryData	Managed Entity ID	**TcAdaptProtMonCurrentData**	**NameGet** (fine grain only)	Attributes from HistoryDataValueType ofCurrentData in Q.822.1
	Period End Time		**HistoryDataValueType: periodEndTime**	
	Suspect Flag		**HistoryDataValueType: suspectIntervalFlag**	
	Number of Suppressed Intervals		**HistoryDataValueType: numSupressedIntevals**	
	No M4 attribute; from Q.822.		**HistoryDataValueType: granularityPeriod**	
	Discarded Cells due to HEC viiolation		NumberDiscCellsHECViolat	
	No actions have been defined.		**getMostRecent** **getBetween**	
	Managed Entity ID		**NameGet** (fine grain only)	
	Performance Parameter and Threshold Value		**counterThresholdListGet** **counterThresholdListSet** **counterThresholdListAdd** **counterThresholdListRemove**	

NOTES for Table 4-1:

1. All attributes are shown in a non-emphasized type font, and CORBA IDL attributes are shown in Section 5 of this document. **Operations are shown in bold type. Logical MIBmanaged Entities and CORBA IDL Objects are shown in bold type.** *Relationships are shown in italics.*

2. The trafficDescriptor logical MIB managed entity covers the full range of Service Categories as defined in the ATM Forum's TM Specification 4.0. In this document's IDL, one interface is used for each Service Category. This approach is taken because

CORBA IDL does not permit a convenient method of describing conditionality. Each of these specialized traffic descriptor interfaces inherits from an uninstantiable parent interface named AtmTrafficDesc that contains common methods related to objectID, profileName, serviceCategory and conformanceDefinition.

5. Information Model IDL

```
#ifndef _atmf_m4nw_v2_idl_
#define _atmf_m4nw_v2_idl_


#include ”itut_x780.idl”
#include ”itut_x780_1.idl”
#include ”itut_m3120.idl”
#include ”itut_q822_1.idl”
#include ”atmf_m4nwconst_v2.idl”


#pragma prefix "atmforum.com"


/**
This IDL code is intended to be stored in a file named "atmf_m4nw_v2.idl"
located in the search path used by IDL compilers on your system.
*/




/**
This module, atmf_m4nw_v2, provides a set of IDL interfaces for managing an
ATM network using the ATM Forum M4 Network View requirements and logical MIB
found in AF-NM-0058.001.  The IDL definitions in this file are object
interfaces. The constant values are contained in a seperate file, but included
in the same "atmf_m4nw_v2" module.
*/
module atmf_m4nw_v2
{




/**
```

5.1. *Imports*

```
*/



/**
Types imported from itut_x780
*/

       typedef itut_x780::AdministrativeStateType AdministrativeStateType;
       typedef itut_x780::AdministrativeStateTypeOpt AdministrativeStateTypeOpt;
       typedef itut_x780::AvailabilityStatusSetType AvailabilityStatusSetType;
       typedef itut_x780::BooleanTypeOpt BooleanTypeOpt;
       typedef itut_x780::DeletePolicyType DeletePolicyType;
       typedef itut_x780::GeneralizedTimeType GeneralizedTimeType;
       typedef itut_x780::Istring Istring;
       typedef itut_x780::IstringSetType IstringSetType;

       typedef itut_x780::MONameType MONameType;
```

```
      typedef itut_x780::MONameSetType MONameSetType;
      typedef itut_x780::NameBindingType NameBindingType;
      typedef itut_x780::ObjectClassType ObjectClassType;
      typedef itut_x780::ObjectClassSetType ObjectClassSetType;
      typedef itut_x780::OperationalStateType OperationalStateType;
      typedef itut_x780::StringSetType StringSetType;
      typedef itut_x780::UIDType UIDType;


/**
Types imported from itut_m3120
*/
      typedef itut_m3120::AccessGroupNameSetType AccessGroupNameSetType;
      typedef itut_m3120::AlarmSeverityAssignmentProfileNameType
      AlarmSeverityAssignmentProfileNameType;
      typedef itut_m3120::ArcProbableCauseSetType ArcProbableCauseSetType;
      typedef itut_m3120::CharacteristicInfoType CharacteristicInfoType;
      typedef itut_m3120::CurrentProblemSetType CurrentProblemSetType;
      typedef itut_m3120::NetworkNameType NetworkNameType;
      typedef itut_m3120::AbstractLinkNameType AbstractLinkNameType;
      typedef itut_m3120::AbstractLinkNameSetType AbstractLinkNameSetType;
      typedef itut_m3120::AbstractLinkEndNameSetType AbstractLinkEndNameSetType;
      typedef itut_m3120::NetworkTPNameSetType NetworkTPNameSetType;
      typedef itut_m3120::ProblemCauseType ProblemCauseType;
      typedef itut_m3120:: PointDirectionalityType PointDirectionalityType;
      typedef itut_m3120:: UsageCostType UsageCostType;
      typedef itut_m3120:: PointCapacityType PointCapacityType;
      typedef itut_m3120:: SubnetworkNameSetType SubnetworkNameSetType;

/**
      SignalId may be set to a single signal id component with characteristic
      Information set to vpCI or vcCI, or may use two signal id components to
      indicate vpCI and vpCI together (e.g., a subnetwork that supports both
      VP and VC subnetwork connections)
      VP Characteristic Information (vpCI=17) and VC Characteristic Information
      (vcCI=18) constants are defined in module itut_m3120:CharacteristicInfoConst
*/
      typedef itut_m3120::SignalIdType SignalIdType;
/**
Types imported from itut_q222_1
*/
      typedef itut_q822d1::TimePeriodType TimePeriodType;


/**
Exceptions imported from x.780

itut_x780::ApplicationError
itut_x780::CreateError
itut_x780::DeleteError

These exceptions all produce an attribute that gives extra details pertaining
to specific error conditions. A few predefined attribute values are listed in
X.780. These values can be used to replace atmf_m4nw exceptions with X.780
exceptions.

Throughout this document, atmf_m4nw exceptions were mapped to the following
x.780 exceptions:
- atmf_m4nw::ObjectsFailure is mapped to x.780::ApplicationError with no
specific error condition.

- atmf_m4nw::OutOfRange is mapped to x.780::ApplicationError with
invalidParameter as a specific error condition.
```

```
- atmf_m4nw::DuplicateName is mapped to x.780::CreateError with duplicateName
as a specific error condition.

- atmf_m4nw::NotSupported is mapped to x.780::CreateError with
unsupportedPackages as a specific error condition during object creation.  Also, if an
operation is not supported, the ApplicationError exception shall be raised with no
specific error condition.

- atmf_m4nw::ContainedObjects is mapped to x.780::DeleteError with
containsObjects as a specific error condition.

- atmf_m4nw::DeleteNotAllowed is mapped to x.780::DeleteError with
notDeletable as a specific error condition.
*/



/**
Interfaces imported from itut_x780

itut_m3120::itut_x780::ManagedObject
itut_m3120::ManagedObjectFactory


Interfaces imported from itut_m3120

itut_m3120::Network
*/




/**
```

5.2. *Forward Declarations*

```
*/


/**
Typedefs forward declarations
*/
       typedef MONameType AtmLinkNameType;
       typedef MONameType AtmLinkEndNameType;
       typedef MONameType AtmLinkEndPhyNameType;
       typedef MONameType AtmNetworkAccessProfileNameType;
       typedef MONameType AtmNetworkCTPNameType;
       typedef MONameType AtmNetworkTTPNameType;
       typedef MONameType AtmRoutingProfileNameType;
       typedef MONameType AtmSNCNameType;
       typedef MONameType AtmSubnetworkNameType;
       typedef MONameType AtmAbstractTrafficDescNameType;
       typedef MONameType AtmTrafficDescABRNameType;
       typedef MONameType AtmTrafficDescCBRNameType;
       typedef MONameType AtmTrafficDescVBRNameType;
       typedef MONameType AtmTrafficDescUBRNameType;
       typedef MONameType AtmTrafficDescGFRNameType;
       typedef MONameType LatestOccurenceLogNameType;
       typedef MONameType AalProfileNameType;
       typedef MONameType CesServiceProfileNameType;
```

```
/**
```

5.3. *Structures and Typedefs*

```
*/


       typedef sequence<AtmNetworkCTPNameType> AtmNetworkCTPNameSeqType;

       typedef sequence<AtmSNCNameType> AtmSNCNameSeqType;

       typedef sequence<AtmSubnetworkNameType> AtmSubnetworkNameSeqType;

/**
Virtual ID - VPI value and VCI value
*/
       struct VirtualIDType
       {
              unsigned longvpi;
              unsigned longvci;
       };

/**
ABR CDF
*/
       enum EnumABRCDFType
       {
              abrCDF_No,          // no CDF
              abrCDF_0,           // 0
              abrCDF_1over64,           // 1/64
              abrCDF_1over32,           // 1/32
              abrCDF_1over16,           // 1/16
              abrCDF_1over8,            // 1/8
              abrCDF_1over4,            // 1/4
              abrCDF_1over2,            // 1/2
              abrCDF_1            // 1
       };

/**
ABR Nrm
*/
       enum EnumABRNrmType
       {
              abrNrm_No,          // no Nrm
              abrNrm_2,           // 2
              abrNrm_4,           // 4
              abrNrm_8,           // 8
              abrNrm_16,          // 16
              abrNrm_32,          // 32
              abrNrm_64,          // 64
              abrNrm_128,         // 128
              abrNrm_256          // 256
       };

/**
ABR Rate Change Factor
*/
       enum EnumABRRateChangeFactorType
       {
```

```
              abrrcf_1over32768,  // 1/32768
              abrrcf_1over16384,  // 1/16384
              abrrcf_1over8192,   // 1/8192
              abrrcf_1over4096,   // 1/4096
              abrrcf_1over2048,   // 1/2048
              abrrcf_1over1024,   // 1/1024
              abrrcf_1over512,    // 1/512
              abrrcf_1over256,    // 1/256
              abrrcf_1over128,    // 1/128
              abrrcf_1over64,           // 1/64
              abrrcf_1over32,           // 1/32
              abrrcf_1over16,           // 1/16
              abrrcf_1over8,            // 1/8
              abrrcf_1over4,            // 1/4
              abrrcf_1over2,            // 1/2
              abrrcf_1            // 1
       };

/**
ABR Trm
*/
       enum EnumABRTrmType
       {
              abrTrm_No,    // no Trm
              abrTrm_1,     // 100 ms
              abrTrm_2,     // 100 * 2^(-1) = 50 ms
              abrTrm_3,     // 100 * 2^(-2) = 25 ms
              abrTrm_4,     // 100 * 2^(-3) = 12.5 ms
              abrTrm_5,     // 100 * 2^(-4) = 6.25 ms
              abrTrm_6,     // 100 * 2^(-5) = 3.125 ms
              abrTrm_7,     // 100 * 2^(-6) = 1.5626 ms
              abrTrm_8      // 100 * 2^(-7) = 0.78125ms
       };

/**
Latest Occurrence Log Definitions
*/
       enum EnumCellHeaderAbnormalityType
       {
              cellHeaderAbnormality_UnassignedVpiVci,
              cellHeaderAbnormality_OutOfRangeVpiVci
       };

/**
TM 4.1 Conformance Definition
*/
       enum EnumConformanceDefinitionType
       {
              conformanceDefinition_Other,
              conformanceDefinition_CBR1,
              conformanceDefinition_VBR1,
              conformanceDefinition_VBR2,
              conformanceDefinition_VBR3,
              conformanceDefinition_UBR1,
              conformanceDefinition_UBR2,
              conformanceDefinition_ABR,
              conformanceDefinition_GFR
       };

/**
A connection type may be broadcast (point-to-multipoint),
merge (multipoint-to-point), composite (root-to-leaves & leaves-to-root),
multipoint (multipoint-to-multipoint), or pointToPoint (point-to-point)
```

```
*/
       enum EnumConnectionType
       {
              Connection_Broadcast,
              Connection_Merge,
              Connection_Composite,
              Connection_Multipoint,
              Connection_PointToPoint
       };

/**
IMA Group State     - See ATM Forum IMA v1.1, Section 10.2.1
*/
       enum EnumImaGroupState
       {
              imaGroupState_notConfigured,
              imaGroupState_startUp,
              imaGroupState_startUpAck,
              imaGroupState_configAbortUnsupportedM,
              imaGroupState_configAbortIncompatibleSymmetry,
              imaGroupState_configAbortOther,
              imaGroupState_insufficientLinks,
              imaGroupState_blocked,
              imaGroupState_operational,
              imaGroupState_configAbortUnsupportedImaVersion
       };

/**
IMA Link State      - See ATM Forum IMA v1.1, Section 10.1.2
*/
       enum EnumImaLinkState
       {
              imaLinkState_notInGroup,
              imaLinkState_unusableNoGivenReason,
              imaLinkState_unusableFault,
              imaLinkState_unusableMisconnected,
              imaLinkState_unusableInhibited,
              imaLinkState_unusableFailed,
              imaLinkState_usable,
              imaLinkState_active
       };

/**
IMA Group Symmetry  - See ATM Forum IMA v1.1, Section 5.2.2.7
*/
       enum EnumImaGroupSymmetry
       {
              imaGroupSymmetry_symmetricOperation,
              imaGroupSymmetry_asymmetricOperation,
              imaGroupSymmetry_asymmetricConfiguration
       };

/**
IMA Group Transmit Clock Mode
Indicate the transmit clock mode of the IMA group.
There are two possible modes: the Common Transmit Clock (CTC) and the Independent
Transmit Clock (ITC).
The CTC mode corresponds to the case when the transmit clock of all IMA links are
derived from the same source.
The ITC configuration corresponds to the case where there is at least one IMA link
whose transmit clock is derived from a source different than at least another link
transmit clock.
*/
```

```
	enum EnumImaGroupTxClkMode
	{
		imaGroupTxClkMode_ctc,
		imaGroupTxClkMode_itc
	};

/**
IMA Frame Length    - See ATM Forum IMA v1.1, Section 5.2.2.4.2
*/
	enum EnumImaFrameLength
	{
		imaFrameLength_m32,
		imaFrameLength_m64,
		imaFrameLength_m128,
		imaFrameLength_m256
	};

/**
IMA Test Procedure Status
The test is started by setting operating status. If any link should fail the test, the
IMA will set the status to linkFail. The linkFail state will persist until either the
disabled state is set or until no instance of imaLinkTestProcStatus has the value
linkFail.
Only the values disabled and operating may be written. Writing the operating value
will not cause clearing of the linkFail state.
*/
	enum EnumImaTestProcStatus
	{
		imaTestProcStatus_disabled,
		imaTestProcStatus_operating,
		imaTestProcStatus_linkFail
	};

	enum EnumAalType
	{
		aalType_aal1,
		aalType_aal2,
		aalType_aal34,
		aalType_aal5
	};

/**
This attribute identifies which AAL1 format should be used. This attribute applies
only to structured format. The default value Basic does not carry channel associated
signalling (CAS) bits and uses a single 125 usec frame. e1Cas, ds1sfCas, and ds1EsfCas
carry CAS bits in a multiframe structure for E1, DS1 SF, and DS1 ESF respectively.
*/

	enum EnumChannelAssociatedSignalling
	{
		casType_basic,
		casType_e1Cas,
		casType_ds1SfCas,
		casType_ds1EsfCas,
		casType_j2Cas
	};


/**
This attribute indicates whether the clock recovery type is Synchronous, SRTS
(Synchronous Residual Time Stamp), or ACR(Adaptive Clock Recovery).
*/
	enum EnumclockRecoveryType
```

```
	{
		clockRecoveryType_synchronous,
		clockRecoveryType_srts,
		clockRecoveryType_adaptive
	};

/**
This attribute indicates the FEC method: no FEC, FEC for Loss Sensitive Signal
Transport, or FEC for Delay Sensitive Signal Transport.
*/

	enum EnumForwardErrorCorrectionMethod
	{
		fecMethod_noFEC,
		fecMethod_lossSensitiveSignalFEC,
		fecMethod_delaySensitiveSignalFEC
	};

/**
This attribute identifies the SSCS type for the AAL. Valid values are NULL, Data SSCS
based on SSCOP (assured operation), Data SSCS based on SSCOP (non-assured operation),
or Frame Relay SSCS.
*/

	enum EnumSscsType
	{
		sscsType_Null,
		sscsType_dataAssured,
		sscsType_dataNonAssured,
		sscsType_frameRelay
	};

/**
This attribute is used to identify the AAL subtype. Valid values for this attribute
are NULL, Voice-band based on 64 kbps, Circuit Emulation (synchronous), Circuit
Emulation (asynchronous), High-quality Audio, and Video.
*/

	enum EnumAal1SubType
	{
		aal1SubType_null,
		aal1SubType_voiceBand,
		aal1SubType_circuitEmulationSynchronous,
		aal1SubType_circuitEmulationAsynchronous,
		aal1SubType_highQualityAudio,
		aal1SubType_video
	};

	enum EnumAal2ConfigResponsibility
	{
		configResp_ilmi,
		configResp_lesEoc,
		configResp_other
	};

	enum EnumSscsSstedStatus
	{
		sscsSstedStatus_selected,
		sscsSstedStatus_notSelected
	};

	enum EnumSscsSsadtStatus
	{
```

```
                sscsSsadtStatus_selected,
                sscsSsadtStatus_notSelected
        };

        enum EnumSscsServiceCategory
        {
                sscsServiceCat_audio,
                sscsServiceCat_multirate,
                sscsServiceCat_audioAndMultiRate
        };

        enum EnumTransportStatus
        {
                transportStatus_disabled,
                transportStatus_enabled
        };

        enum EnumSscsProfileSource
        {
                sscsProfSource_itut,
                sscsProfSource_other
        };

        enum EnumSscsPcmEncoding
        {
                sscsPcmEncoding_aLaw,
                sscsPcmEncoding_uLaw
        };

        enum EnumCpsOptimisation
        {
                cpsOptimisation_singleCpsPacketPerCpsPduNoOverlap,
                cpsOptimisation_multipleCpsPacketsPerCpsPduWithOverlap
        };


        enum EnumAalMode
        {
                aalMode_messageAssured,
                aalMode_messageUnassured,
                aalMode_streamingAssured,
                aalMode_streamingUnassured
        };


/**
Element resulting from a connection trace
*/
        struct ConnTraceType
        {
                MONameType    linkOrLinkEnd;
                VirtualIDType       virtualID;
        };

/**
List of connection trace results
*/
        typedef sequence<ConnTraceType> ConnTraceSeqType;

/**
GFR1 or GFR2
*/
        enum EnumGFR1or2Type
```

```
	{
		gfr1or2_1,
		gfr1or2_2
	};

/**
Latest Occurrence Log Entry
*/
	struct LatestOccurrenceLogEntryType
	{
		AtmLinkEndNameType		linkEnd;
		VirtualIDType				virtualID;
		EnumCellHeaderAbnormalityType	abnormalityType;
		GeneralizedTimeType		timeStamp;
	};

/**
List of Latest Occurrence Log Entries
*/
	typedef sequence<LatestOccurrenceLogEntryType>
		LatestOccurrenceLogEntrySetType;

/**
Link End type
*/
	enum EnumLinkEndType
	{
		LinkEnd_Uni,
		LinkEnd_IntraNNI,
		LinkEnd_InterNNI,
		LinkEnd_Pnni,
		LinkEnd_Unconfigured
	};



/**
Type of Link trace request
*/
	enum EnumLinkTraceType
	{
		LinkTrace_AllSubnets,
		LinkTrace_AllInVPLND,
		LinkTrace_AllInVCLND,
		LinkTrace_SelectedSubnets
	};


/**
Results of loopback request
*/
	struct LoopbackCellReplyType
	{
		boolean			success;
		ProblemCauseType	problemCause;
	};

/**
Loopback location code
*/
	typedef sequence<octet>	LoopbackLocationCodeSeqType;

/**
```

```
Loopback Location
*/
       struct LoopbackLocType
       {
              boolean                           endPoint;
              LoopbackLocationCodeSeqType       locationCode;
       };


/**
PM OAM block size
*/
       enum EnumPmOamBlockSizeType
       {
              pmOamBlockSize_128,
              pmOamBlockSize_256,
              pmOamBlockSize_512,
              pmOamBlockSize_1024
       };

/**
PM OAM cell type for loopback
*/
       enum EnumOamCellType
       {
              oamCell_Segment,
              oamCell_EndToEnd
       };

/**
OAM loopback direction
*/

       enum EnumOamLoopbackDirectionType
       {
              oamLoopbackDirection_Receive,
              oamLoopbackDirection_Transmit
       };

/**
PM OAM direction
*/
       enum EnumPmOamDirectionType
       {
              pmOamDirection_Receive,
              pmOamDirection_Transmit,
              pmOamDirection_Both
       };

/**
PM OAM method
*/
       enum EnumPmOamMethodType
       {
              pmOamMethod_TMN,
              pmOamMethod_OAM,
              pmOamMethod_NotSupported
       };

/**
Port ID, managed element and port required, others optional
*/
       struct PortIDType
```

```
	{
		Istring		managedElement;
		Istring		bay;
		Istring		shelf;
		Istring		drawer;
		Istring		slot;
		Istring		port;
	};

/**
Provision type may be manual or automatic
*/
	enum EnumProvisionType
	{
		provision_Manual,
		provision_Automatic
	};

/**
restoration mode
*/
	enum EnumRestorationModeType
	{
		restorationMode_Unavailable,
		restorationMode_AvailRoutingOnly,
		restorationMode_AvailReRoutingOnly,
		restorationMode_AvailRoutingAndReRouting
	};

	enum EnumRoutingOptionType
	{
		routingOption_Mandatory,
			// must use the object in establishing the connection
		routingOption_Preferred,
			// attempt to use the object in establishing the connection
		routingOption_Exclude,
			// do not use the object in establishing connection
		routingOption_sameRoute,
			// use same route as referenced object
		routingOption_diverseRoute
			//use different route than referenced object
	};

	struct RouteDescriptionType
	{
		MONameType		referenceObject;
		EnumRoutingOptionType	option;
	};

	typedef sequence<RouteDescriptionType> RouteDescriptionListType;



/**
Traffic Service Category
*/
	enum EnumServiceCategoryType
	{
		serviceCategory_Other,
		serviceCategory_CBR,
		serviceCategory_RtVBR,
		serviceCategory_NrtVBR,
		serviceCategory_ABR,
```

```
              serviceCategory_UBR,
              serviceCategory_GFR
       };

/**
VPI or VCI range
*/
       struct VpiOrVciRangeType
       {
              long   low;
              long   high;
       };

/**
Description of Z-end TP (CTP) for multipoint request
*/
       struct ZTPCompositeCTPType
       {
              AtmNetworkCTPNameType            ztp;
              boolean                          ztpTrailEndPointInd;
              AtmAbstractTrafficDescNameType   ztpEgressTrafficDescProfile;
       };

/**
List of Z-end TP (CTP) descriptions for multipoint request
*/
       typedef sequence<ZTPCompositeCTPType> ZTPCompositeCtpSetType;

/**
Description of Z-end TP (LinkEnd) for multipoint request
*/
       struct ZTPCompositeLinkEndType
       {
              AtmLinkEndNameType         ztp;
              VirtualIDType                    ztpVirtualID;
              boolean                          ztpTrailEndPointInd;
              AtmAbstractTrafficDescNameType   ztpEgressTrafficDescProfile;
       };

/**
List of Z-end TP (LinkEnd) descriptions for multipoint request
*/
       typedef sequence<ZTPCompositeLinkEndType> ZTPCompositeLinkEndSetType;
```

```
/**
```

5.4. *Intefaces – Facade*

```
The behaviour of the facade interfaces are identical to the corresponding
fine-grained interfaces.

This section can be omitted from IDL if a management system only supports
fine-grained interfaces.
*/

/**
All façade objects inherit the following from X.780:ManagedObject_F:
       objectClassGet
       packagesGet
       creationSourceGet
       deletePolicyGet
       attributesGet
       attributesBulkGet
       destroy
**/


/**
```

5.4.1 AbstractAalProfile

```
*/

/**    The aalProfile object class is a managed support object used to organize data
       that describes the AAL processing functions of the ATM NE. The attribute
       aalType identifies the type of AAL processing (i.e., AAL1, AAL3/4, or AAL5).

       The AAL profiling information is contained in subclasses of AbstractAalProfile
       which are present based on the value of the aalType attribute.
*/

       valuetype AbstractAalProfileValueType: itut_x780::ManagedObjectValueType
       {
              public EnumAalType  aalType;
                     // GET, SET-BY-CREATE

       }; // valuetype AbstractAalProfileValueType


       interface AbstractAalProfile_F: itut_x780::ManagedObject_F
       {
/**    This attribute identifies the AAL Type. Valid types are AAL1, AAL3/4, and AAL5.
*/
              EnumAalType  aalTypeGet(in MONameType name)
                     raises (itut_x780::ApplicationError);

              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, objectCreation)
              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, objectDeletion)

       }; // interface AbstractAalProfile_F


/**
```

5.4.2 AalProfileTypeOne

```
*/

      valuetype AalProfileTypeOneValueType: AbstractAalProfileValueType
      {
            public long  cbrRate;
                  // GET, SET-BY-CREATE
            public long  cellLossIntegrationPeriod;
                  // GET, SET-BY-CREATE
            public EnumclockRecoveryType     clockRecoveryType;
                  // GET, SET-BY-CREATE
            public EnumForwardErrorCorrectionMethod forwardErrorCorrectionMethod;
                  // GET, SET-BY-CREATE
            public long  partiallyFilledCells;
                  // GET, SET-BY-CREATE
            public boolean      structuredDataTransfer;
                  // GET, SET-BY-CREATE
            public EnumAal1SubType    subType;
                  // GET, SET-BY-CREATE

      }; // valuetype AalProfileTypeOneValueType

      interface AalProfileTypeOne_F: AbstractAalProfile_F

      {
/**   This attribute represents the rate of the CBR service supported by the AAL.
*/
            long   cbrRateGet(in MONameType name)
                  raises (itut_x780::ApplicationError);

/**   This attribute identifies the time in milliseconds for the cell loss
      integration period. If cells are lost for this period of time, the containing
      AtmInterworkingTTP object will generate a communications alarm.
*/
            long   cellLossIntegrationPeriodGet(in MONameType name)
                  raises (itut_x780::ApplicationError);

/**   This attribute indicates whether the clock recovery type is Synchronous, SRTS
      (Synchronous Residual Time Stamp), or Adaptive Clock Recovery.
*/
            EnumclockRecoveryType clockRecoveryTypeGet(in MONameType name)
                  raises (itut_x780::ApplicationError);

/**   This attribute indicates the FEC method: no FEC, FEC for Loss Sensitive Signal
      Transport, or FEC for Delay Sensitive Signal Transport.
*/

            EnumForwardErrorCorrectionMethod forwardErrorCorrectionMethodGet
                  (in MONameType name)
                  raises (itut_x780::ApplicationError);

            long   partiallyFilledCellsGet(in MONameType name)
                  raises (itut_x780::ApplicationError);

            boolean      structuredDataTransferGet(in MONameType name)
                  raises (itut_x780::ApplicationError);

            EnumAal1SubType     subTypeGet(in MONameType name)
                  raises (itut_x780::ApplicationError);


            MANDATORY_NOTIFICATION(
```

```
                itut_x780::Notifications, objectCreation)
        MANDATORY_NOTIFICATION(
                itut_x780::Notifications, objectDeletion)

    }; // interface AalProfileTypeOne_F

    interface AalProfileTypeOneFactory: itut_x780::ManagedObjectFactory
    {
        itut_x780::ManagedObject create
            (in NameBindingType nameBinding,
                // module name containing Name Binding info.
            in MONameType superior,
                // Name of containing object.
            in string reqID,
                // Requested ID value for name, will be
                // empty if auto-naming is to be used.
            out MONameType name,
                // Entire name of newly created object.
            in StringSetType packageNameList,
                // List of packages requested.

            in long      cbrRate,
            in long      cellLossIntegrationPeriod,
            in EnumclockRecoveryType   clockRecoveryType,
            in EnumForwardErrorCorrectionMethod
                forwardErrorCorrectionMethod,
            in long      partiallyFilledCells,
            in boolean   structuredDataTransfer,
            in EnumAal1SubType  subType)

            raises (itut_x780::ApplicationError,
                itut_x780::CreateError);

    }; // interface AalProfileTypeOneFactory

/**
```

5.4.3 AalProfileTypeTwo

```
*/

valuetype AalProfileTypeTwoValueType: AbstractAalProfileValueType
    {
        public long  applicationIdentifier;
            // GET, SET-BY-CREATE
        public EnumAal2ConfigResponsibility     configResponsibility;
            // GET, SET-BY-CREATE
        public long  cpsMaxMultiplexedChannels;
            // GET, SET-BY-CREATE
        public long  cpsMaxSduLength;
            // GET, SET-BY-CREATE
        public long  sscsMaxSssarSduLength;
            // GET, SET-BY-CREATE
        public EnumSscsSstedStatus sscsSstedStatus;
            // GET, SET-BY-CREATE
        public EnumSscsSsadtStatus sscsSsadtStatus;
            // GET, SET-BY-CREATE
        public EnumSscsServiceCategory sscsServiceCategory;
            // GET, SET-BY-CREATE
        public EnumTransportStatus sscsAudioServiceTransport;
            // GET, SET-BY-CREATE
        public EnumSscsProfileSource sscsProfileSource;
            // GET, SET-BY-CREATE
```

```
            public long  sscsIeeeOui;
                  // GET, SET-BY-CREATE
            public long  SscsPredefinedProfileIdentifier;
                  // GET, SET-BY-CREATE
            public EnumSscsPcmEncoding sscsPcmEncoding;
                  // GET, SET-BY-CREATE
            public EnumTransportStatus sscsFaxDemodulationTransport;
                  // GET, SET-BY-CREATE
            public EnumTransportStatus sscsCasSignalingTransport;
                  // GET, SET-BY-CREATE
            public EnumTransportStatus sscsDtmfDigitPacketTransport;
                  // GET, SET-BY-CREATE
            public EnumTransportStatus sscsMfR1DigitPacketTransport;
                  // GET, SET-BY-CREATE
            public EnumTransportStatus sscsMfR2DigitPacketTransport;
                  // GET, SET-BY-CREATE
            public EnumTransportStatus sscsCircuitModeDataTransport;
                  // GET, SET-BY-CREATE
            public long  sscsCircuitModeDataNumChannels;
                  // GET, SET-BY-CREATE
            public EnumTransportStatus sscsFrameModeDataTransport;
                  // GET, SET-BY-CREATE
            public long  sscsFrameModeDataMaxLength;
                  // GET, SET-BY-CREATE
            public long  sscopSduLength;
                  // GET, SET-BY-CREATE
            public long  sscopUuLength;
                  // GET, SET-BY-CREATE
      }; // valuetype AalProfileTypeTwoValueType

      interface AalProfileTypeTwo_F: AbstractAalProfile_F
      {

/**   The application identifier specifies which AAL2 application is using the PVC.
      In addition it specifies protocol combinations to be used on AAL2 channels
      between IWFs, i.e. it implicitly configures the use of I.366.1 or I.366.2 for
      bearer and signalling AAL2 channels. The value of this object should be one of
      those values given in the administered list of AppIds for AAL2 Applications in
      section 5 of the ATM Forum document of well-known addresses and assigned codes
      which can be located at
      http://www.atmforum.com/atmforum/specs/public_assigned_codes.pdf
*/
            long applicationIdentifierGet(in MONameType name)
                  raises (itut_x780::ApplicationError);

/**   This object identifies which management channel is responsible for AAL2
      provisioning and auto-configuration. The user-side IME must check the value of
      this parameter first before configuring the AAL2 for the PVC to determine
      whether or not this responsibility lies with ILMI.

      ilmi – if this is selected then the user-side IME must use the AAL2 Profile
      defined in this entry and associated AAL2 Profile extension table entry to
      auto-configure the AAL2 for the PVC.

      lesEoc – if this is selected then the user-side IME must not auto-configure the
      AAL2 using ILMI. The AAL2 Profile parameters do not need to be provisioned by
      the network-side IME and with the exception of this object the AAL2 profile
      parameters in the ILMI MIB are not applicable. The Loop Emulation Service
      Embedded Operations Channel (LES EOC) shall be used to provision the AAL2 for
      the PVC. The start-up AAL2 parameters for this PVC must assume the default
      values specified in the af-vmoa-0145.000 MIB such that the LES EOC is
      operational.
```

```
      other – if this is selected then the user-side IME must not auto-configure the
      AAL2 using ILMI. The AAL2 Profile parameters do not need to be provisioned by
      the network-side IME and with the exception of this object the AAL2 profile
      parameters in the ILMI MIB are not applicable. Another mechanism will be used
      to provision the AAL2 at the NT for the PVC.
*/
            EnumAal2ConfigResponsibility     configResponsibilityGet
                  (in MONameType name)
                  raises (itut_x780::ApplicationError);

            long   cpsMaxMultiplexedChannelsGet(in MONameType name)
                  raises (itut_x780::ApplicationError);

/**   Maximum CPS-SDU size given in octets. This parameter is bidirectional, i.e. its
      value applies to both directions of the AAL2 connection.
*/
            long   cpsMaxSduLengthGet(in MONameType name)
                  raises (itut_x780::ApplicationError);

/**   This is the maximum size and SSSAR-SDU can take as defined in I.366.1. This
      parameter only applies to those AAL2 channels using I.366.1 SSSAR. This
      parameter is bidirectional, i.e its value applies to both directions of the
      AAL2 connection. Note that if the appID indicates Loop Emulation Service as the
      AAL2 application then the minimum allowable value of this parameter is 493.
*/
            long   sscsMaxSssarSduLengthGet(in MONameType name)
                  raises (itut_x780::ApplicationError);

/**   This attribute specifies whether the SSTED service is used or not. If the value
      of this object is 'selected' then I.366.1 SSSAR must be supported by the AAL2.
      This parameter only applies to those AAL2 channels using I.366.1 SSSAR.
*/
            EnumSscsSstedStatus sscsSstedStatusGet(in MONameType name)
                  raises (itut_x780::ApplicationError);

/**   This attribute specifies whether SSADT service is used or not. Note that if the
      value of this object is 'selected' then I.366.1 SSSAR must be supported and by
      implication the SSTED service must also be 'selected'. This parameter only
      applies to those AAL2 channels using I.366.1 SSSAR.
*/
            EnumSscsSsadtStatus sscsSsadtStatusGet(in MONameType name)
                  raises (itut_x780::ApplicationError);

/**   This attribute describes the type of service category supported by this
      AAL2 PVC.
*/
            EnumSscsServiceCategory sscsServiceCategoryGet(in MONameType name)
                  raises (itut_x780::ApplicationError);

/**   This attribute describes whether the Audio Service that is used for
      transporting voice and voiceband data is enabled or disabled.
*/
            EnumTransportStatus sscsAudioServiceTransportGet
                  (in MONameType name)
                  raises (itut_x780::ApplicationError);


            EnumSscsProfileSource sscsProfileSourceGet(in MONameType name)
                  raises (itut_x780::ApplicationError);

/**   This attribute contains the IEEE Organizationally Unique Identifier (OUI) of
      the organization that specified the profile being used, if other than ITU-T.
            For example, if the source is the ATM Forum, the value of this object is
```

```
            00A03E. This attribute is only meaningful if SccsProfileSource has the
            value 'other'.
*/
            long   sscsIeeeOuiGet(in MONameType name)
                   raises (itut_x780::ApplicationError);

            long   SscsPredefinedProfileIdentifierGet(in MONameType name)
                   raises (itut_x780::ApplicationError);

/**    This attribute describes the type of PCM encoding used.
*/
            EnumSscsPcmEncoding sscsPcmEncodingGet(in MONameType name)
                   raises (itut_x780::ApplicationError);

/**    This attribute describes whether transport of demodulated facsimile data is
       enabled or disabled.
*/
            EnumTransportStatus sscsFaxDemodulationTransportGet
                   (in MONameType name)
                   raises (itut_x780::ApplicationError);

/**    This attribute describes whether transport of CAS signaling bits is enabled or
       disabled.
*/
            EnumTransportStatus sscsCasSignalingTransportGet
                   (in MONameType name)
                   raises (itut_x780::ApplicationError);

            EnumTransportStatus sscsDtmfDigitPacketTransportGet
                   (in MONameType name)
                   raises (itut_x780::ApplicationError);


            EnumTransportStatus sscsMfR1DigitPacketTransportGet
                   (in MONameType name)
                   raises (itut_x780::ApplicationError);


            EnumTransportStatus sscsMfR2DigitPacketTransportGet
                   (in MONameType name)
                   raises (itut_x780::ApplicationError);

            EnumTransportStatus sscsCircuitModeDataTransportGet
                   (in MONameType name)
                   raises (itut_x780::ApplicationError);

/**    This attribute describes the multiplier N in N*64kbit/s circuit mode data. It
       is only applicable if the value of sscsCircuitModeDataTransport is 'enabled'.
*/
            long   sscsCircuitModeDataNumChannelsGet(in MONameType name)
                   raises (itut_x780::ApplicationError);

            EnumTransportStatus sscsFrameModeDataTransportGet
                   (in MONameType name)
                   raises (itut_x780::ApplicationError);

/**    This attribute specifies the maximum length of I.366.2 frame mode data. It is
       only applicable if the value of sscsFrameModeDataTransport is 'enabled'.
*/
            long   sscsFrameModeDataMaxLengthGet(in MONameType name)
                   raises (itut_x780::ApplicationError);

/**    The Maximum SSCOP-SDU length. This attribute is only required is the value of
```

```
      sscsSsadtStatus is selected else it is not applicable.
*/
            long   SscopSduLengthGet(in MONameType name)
                   raises (itut_x780::ApplicationError);

/**   The Maximum SSCOP-UU field length. This attribute is only required is the value
      of sscsSsadtStatus is selected else it is not applicable.
*/
            long   sscopUuLengthGet(in MONameType name)
                   raises (itut_x780::ApplicationError);


            MANDATORY_NOTIFICATION(
                   itut_x780::Notifications, objectCreation)
            MANDATORY_NOTIFICATION(
                   itut_x780::Notifications, objectDeletion)

      }; // interface AalProfileTypeTwo_F

      interface AalProfileTypeTwoFactory: itut_x780::ManagedObjectFactory
      {
            itut_x780::ManagedObject create
                   (in NameBindingType nameBinding,
                          // module name containing Name Binding info.
                   in MONameType superior,
                          // Name of containing object.
                   in string reqID,
                          // Requested ID value for name, will be
                          // empty if auto-naming is to be used.
                   out MONameType name,
                          // Entire name of newly created object.
                   in StringSetType packageNameList,
                          // List of packages requested.

                   in long      applicationIdentifier,
                   in EnumAal2ConfigResponsibility  configResponsibility,
                   in long      cpsMaxMultiplexedChannels,
                   in long      cpsMaxSduLength,
                   in long      sscsMaxSssarSduLength,
                   in EnumSscsSstedStatus sscsSstedStatus,
                   in EnumSscsSsadtStatus sscsSsadtStatus,
                   in EnumSscsServiceCategory sscsServiceCategory,
                   in EnumTransportStatus sscsAudioServiceTransport,
                   in EnumSscsProfileSource sscsProfileSource,
                   in long      sscsIeeeOui,
                   in long      SscsPredefinedProfileIdentifier,
                   in EnumSscsPcmEncoding sscsPcmEncoding,
                   in EnumTransportStatus sscsFaxDemodulationTransport,
                   in EnumTransportStatus sscsCasSignalingTransport,
                   in EnumTransportStatus     sscsDtmfDigitPacketTransport,
                   in EnumTransportStatus     sscsMfR1DigitPacketTransport,
                   in EnumTransportStatus     sscsMfR2DigitPacketTransport,
                   in EnumTransportStatus     sscsCircuitModeDataTransport,
                   in long      sscsCircuitModeDataNumChannels,
                   in EnumTransportStatus     sscsFrameModeDataTransport,
                   in long      sscsFrameModeDataMaxLength,
                   in long      sscopSduLength,
                   in long      sscopUuLength)

                   raises (itut_x780::ApplicationError,
                          itut_x780::CreateError);

      }; // interface AalProfileTypeTwoFactory
```

```
/**
```

5.4.4 AalProfileTypeTwoTrunking

```
*/

/**    The AalProfileTypeTwoTrunking interface provides AAL2 parameters of operation
       for ATM Trunking applications to be used in addition to those provided in the
       Common AAL2 Profile interface. It provides complete configuration of AAL2 for
       PVCs that support ATM Trunking using AAl2 as defined in af-vtoa-0113.000.
*/

valuetype AalProfileTypeTwoTrunkingValueType: AalProfileTypeTwoValueType
       {
              public long  vcci;
                     // GET, SET-BY-CREATE
              public long  signalingVcci;
                     // GET, SET-BY-CREATE
       }; // valuetype AalProfileTypeTwoTrunkingValueType


interface AalProfileTypeTwoTrunking_F: AalProfileTypeTwo_F
       {

/**    This attribute uniquely identifies a PVC between IWFs.
*/
              long   vcciGet(in MONameType name)
                     raises (itut_x780::ApplicationError);
/**    This attribute specifies the VCCI of the PVC that is used to carry the CCS for
       this PVC.
*/
              long   signalingVcciGet(in MONameType name)
                     raises (itut_x780::ApplicationError);

              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, objectCreation)
              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, objectDeletion)

       }; // interface AalProfileTypeTwoTrunking_F

       interface AalProfileTypeTwoTrunkingFactory:
              itut_x780::ManagedObjectFactory
       {
              itut_x780::ManagedObject create
                     (in NameBindingType nameBinding,
                            // module name containing Name Binding info.
                     in MONameType superior,
                            // Name of containing object.
                     in string reqID,
                            // Requested ID value for name, will be
                            // empty if auto-naming is to be used.
                     out MONameType name,
                            // Entire name of newly created object.
                     in StringSetType packageNameList,
                            // List of packages requested.

                     in long      applicationIdentifier,
                     in EnumAal2ConfigResponsibility  configResponsibility,
                     in long      cpsMaxMultiplexedChannels,
                     in long      cpsMaxSduLength,
                     in long      sscsMaxSssarSduLength,
```

```
                    in EnumSscsSstedStatus sscsSstedStatus,
                    in EnumSscsSsadtStatus sscsSsadtStatus,
                    in EnumSscsServiceCategory sscsServiceCategory,
                    in EnumTransportStatus sscsAudioServiceTransport,
                    in EnumSscsProfileSource sscsProfileSource,
                    in long      sscsIeeeOui,
                    in long      SscsPredefinedProfileIdentifier,
                    in EnumSscsPcmEncoding sscsPcmEncoding,
                    in EnumTransportStatus sscsFaxDemodulationTransport,
                    in EnumTransportStatus sscsCasSignalingTransport,
                    in EnumTransportStatus     sscsDtmfDigitPacketTransport,
                    in EnumTransportStatus     sscsMfR1DigitPacketTransport,
                    in EnumTransportStatus     sscsMfR2DigitPacketTransport,
                    in EnumTransportStatus     sscsCircuitModeDataTransport,
                    in long      sscsCircuitModeDataNumChannels,
                    in EnumTransportStatus     sscsFrameModeDataTransport,
                    in long      sscsFrameModeDataMaxLength,
                    in long      sscopSduLength,
                    in long      sscopUuLength,
                    in long      vcci,
                    in long      signalingVcci)

                    raises (itut_x780::ApplicationError,
                           itut_x780::CreateError);

      }; // interface AalProfileTypeTwoTrunkingFactory

/**
```

5.4.5 AalProfileTypeTwoLES

```
*/

/**   An AalProfileTypeTwoLES interface provides AAL2 parameters of operation for LES
      applications to be used in extension to those provided in the Common AAL2
      Profile interface. It provides complete configuration of AAL2 for PVCs that
      support LES using AAl2 as defined in af-vmoa-0145.000."
*/

valuetype AalProfileTypeTwoLESValueType: AalProfileTypeTwoValueType
      {
            public long  cpsCIDLowerLimit;
                  // GET, SET-BY-CREATE
            public long  cpsCIDUpperLimit;
                  // GET, SET-BY-CREATE
            public EnumCpsOptimisation cpsOptimisation;
                  // GET, SET-BY-CREATE
      }; // valuetype AalProfileTypeTwoLESValueType


interface AalProfileTypeTwoLES_F: AalProfileTypeTwo_F
      {

/**   This attribute specifies the minimum value the CID can take for AAL2 bearer
      channels and ISDN-D channels.
*/
            long   cpsCIDLowerLimitGet(in MONameType name)
                  raises (itut_x780::ApplicationError);

/**   This attribute specifies the maximum value the CID can take for AAL2 bearer
      channels and ISDN-D channels.
*/
            long   cpsCIDUpperLimitGet (in MONameType name)
```

```
                    raises (itut_x780::ApplicationError);

/**    This attribute refers to the mode of operation of the CPS on the CP-IWF The
       value of this object imposes a restriction on the CPS-SDU length and hence must
       override the value of CpsMaxSDULength.

       singleCpsPacketPerCpsPduNoOverlap - A single CPS Packet is contained within a
       CPS-PDU and no overlap can occur into the next CPS-PDU. If this option is
       selected, then Timer_CU is not applicable. Also Max CPS-SDU size must be less
       than or equal to 44.

       multipleCpsPacketsPerCpsPduWithOverlap - Multiple CPS Packets are contained
       within a CPS-PDU and overlap can occur into the next CPS-PDU. If this option is
       selected, then Timer_CU is applicable. Also Max CPS-SDU size must be less than
       or equal to 64
*/

              EnumCpsOptimisation cpsOptimisationGet(in MONameType name)
                    raises (itut_x780::ApplicationError);

              MANDATORY_NOTIFICATION(
                    itut_x780::Notifications, objectCreation)
              MANDATORY_NOTIFICATION(
                    itut_x780::Notifications, objectDeletion)

       }; // interface AalProfileTypeTwoLES_F

       interface AalProfileTypeTwoLESFactory: itut_x780::ManagedObjectFactory
       {
              itut_x780::ManagedObject create
                    (in NameBindingType nameBinding,
                          // module name containing Name Binding info.
                    in MONameType superior,
                          // Name of containing object.
                    in string reqID,
                          // Requested ID value for name, will be
                          // empty if auto-naming is to be used.
                    out MONameType name,
                          // Entire name of newly created object.
                    in StringSetType packageNameList,
                          // List of packages requested.

                    in long      applicationIdentifier,
                    in EnumAal2ConfigResponsibility  configResponsibility,
                    in long      cpsMaxMultiplexedChannels,
                    in long      cpsMaxSduLength,
                    in long      sscsMaxSssarSduLength,
                    in EnumSscsSstedStatus sscsSstedStatus,
                    in EnumSscsSsadtStatus sscsSsadtStatus,
                    in EnumSscsServiceCategory sscsServiceCategory,
                    in EnumTransportStatus sscsAudioServiceTransport,
                    in EnumSscsProfileSource sscsProfileSource,
                    in long      sscsIeeeOui,
                    in long      SscsPredefinedProfileIdentifier,
                    in EnumSscsPcmEncoding sscsPcmEncoding,
                    in EnumTransportStatus sscsFaxDemodulationTransport,
                    in EnumTransportStatus sscsCasSignalingTransport,
                    in EnumTransportStatus     sscsDtmfDigitPacketTransport,
                    in EnumTransportStatus     sscsMfR1DigitPacketTransport,
                    in EnumTransportStatus     sscsMfR2DigitPacketTransport,
                    in EnumTransportStatus     sscsCircuitModeDataTransport,
                    in long      sscsCircuitModeDataNumChannels,
                    in EnumTransportStatus     sscsFrameModeDataTransport,
```

```
                    in long      sscsFrameModeDataMaxLength,
                    in long      sscopSduLength,
                    in long      sscopUuLength,
                    in long      cpsCIDLowerLimit,
                    in long      cpsCIDUpperLimit,
                    in EnumCpsOptimisation cpsOptimisation)

                    raises (itut_x780::ApplicationError,
                          itut_x780::CreateError);

      }; // interface AalProfileTypeTwoLESFactory



/**
```

5.4.6 AalProfileTypeThreeFour

```
*/


valuetype AalProfileTypeThreeFourValueType: AbstractAalProfileValueType
      {
            public long  maxCpcsSduSizeForward;
                  // GET, SET-BY-CREATE
            public long  maxCpcsSduSizeBackward;
                  // GET, SET-BY-CREATE
            public long  midRangeLow;
                  // GET, SET-BY-CREATE
            public long  midRangeHigh;
                  // GET, SET-BY-CREATE
            public EnumAalMode aalMode;
                  // GET, SET-BY-CREATE
            public EnumSscsType sscsType;
                  // GET, SET-BY-CREATE

      }; // valuetype AalProfileTypeThreeFourValueType


      interface AalProfileTypeThreeFour_F: AbstractAalProfile_F

      {
/**   These attributes represents the maximum CPCS_PDU size that will be transmitted
      over the connection in both the incoming (forward) and outgoing (backward)
      direction of transmission.
*/
            long   maxCpcsSduSizeForwardGet(in MONameType name)
                  raises (itut_x780::ApplicationError);
            long   maxCpcsSduSizeBackwardGet(in MONameType name)
                  raises (itut_x780::ApplicationError);


            long   midRangeLowGet(in MONameType name)
                  raises (itut_x780::ApplicationError);
            long   midRangeHighGet(in MONameType name)
                  raises (itut_x780::ApplicationError);

/**   This attribute indicates whether the AAL for the supporting VCC is operating in
      message mode or streaming mode, assured or unassured.
*/
            EnumAalMode aalModeGet(in MONameType name)
                  raises (itut_x780::ApplicationError);
```

```
            EnumSscsType  sscsTypeGet(in MONameType name)
                  raises (itut_x780::ApplicationError);

            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectCreation)
            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectDeletion)

      }; // interface AalProfileTypeThreeFour_F


      interface AalProfileTypeThreeFourFactory: itut_x780::ManagedObjectFactory
      {
            itut_x780::ManagedObject create
                  (in NameBindingType nameBinding,
                        // module name containing Name Binding info.
                  in MONameType superior,
                        // Name of containing object.
                  in string reqID,
                        // Requested ID value for name, will be
                        // empty if auto-naming is to be used.
                  out MONameType name,
                        // Entire name of newly created object.
                  in StringSetType packageNameList,
                        // List of packages requested.

                  in long      maxCpcsSduSizeForward,
                  in long      maxCpcsSduSizeBackward,
                  in long            midRangeLow,
                  in long            midRangeHigh,
                  in EnumAalMode     aalMode,
                  in EnumSscsType    sscsType)

                  raises (itut_x780::ApplicationError,
                        itut_x780::CreateError);

      }; // interface AalProfileTypeThreeFourFactory



/**
```

5.4.7 AalProfileTypeFive

```
*/
      valuetype AalProfileTypeFiveValueType: AbstractAalProfileValueType
      {
            public long  maxCpcsSduSizeForward;
                  // GET, SET-BY-CREATE
            public long  maxCpcsSduSizeBackward;
                  // GET, SET-BY-CREATE
            public EnumAalMode  aalMode;
                  // GET, SET-BY-CREATE
            public EnumSscsType sscsType;
                  // GET, SET-BY-CREATE

      }; // valuetype AalProfileTypeFiveValueType

      interface AalProfileTypeFive_F: AbstractAalProfile_F

      {
/**   This multi-valued attribute represents the maximum CPCS_PDU size that will be
```

```
      transmitted over the connection in both the incoming (forward) and outgoing
      (backward) direction of transmission.
*/
            long   maxCpcsSduSizeForwardGet(in MONameType name)
                   raises (itut_x780::ApplicationError);
            long   maxCpcsSduSizeBackwardGet(in MONameType name)
                   raises (itut_x780::ApplicationError);

/**   This attribute indicates whether the AAL for the supporting VCC is operating in
      message mode or streaming mode, assured or unassured.
*/
            EnumAalMode aalModeGet(in MONameType name)
                   raises (itut_x780::ApplicationError);

            EnumSscsType  sscsTypeGet(in MONameType name)
                   raises (itut_x780::ApplicationError);

            MANDATORY_NOTIFICATION(
                   itut_x780::Notifications, objectCreation)
            MANDATORY_NOTIFICATION(
                   itut_x780::Notifications, objectDeletion)

      }; // interface AalProfileTypeFive_F

      interface AalProfileTypeFiveFactory: itut_x780::ManagedObjectFactory
      {
            itut_x780::ManagedObject create
                   (in NameBindingType nameBinding,
                          // module name containing Name Binding info.
                   in MONameType superior,
                          // Name of containing object.
                   in string reqID,
                          // Requested ID value for name, will be
                          // empty if auto-naming is to be used.
                   out MONameType name,
                          // Entire name of newly created object.
                   in StringSetType packageNameList,
                          // List of packages requested.

                   in long      maxCpcsSduSizeForward,
                   in long      maxCpcsSduSizeBackward,
                   in EnumAalMode      aalMode,
                   in EnumSscsType     sscsType)

                   raises (itut_x780::ApplicationError,
                          itut_x780::CreateError);

      }; // interface AalProfileTypeFiveFactory



/**
```

5.4.8 AtmImaGroupTP_F

```
*/
      valuetype AtmImaGroupTPValueType: itut_m3120::NetworkTPValueType
      {

      /**
      AtmImaGroupTPValueType uses the following inherited from
      itut_m3120:: TP:
            public OperationalStateType            operationalState;
```

```
                    // conditional, present if an instance supports it.
                    // GET
              public AlarmStatusType                  alarmStatus;
                    // conditional, present if the TP supports communications
                    // alarm notification.
                    // GET
              public CurrentProblemSetType            currentProblemList;
                    // conditional, present if the TP supports communications
                    // alarm notification.
                    // GET
              public AlarmSeverityAssignmentProfileNameType
                    alarmSeverityAssignmentProfilePointer;
                    // conditional, present if an instance supports
                    // configuration of alarm severities.
                    // GET-REPLACE


       AtmImaGroupTPValueType uses the following inherited from
       itut_m3120:: NetworkTP:
              public PointDirectionalityType   pointDirectionality;
                    // GET
              public SignalIdType        signalId;
                    // GET, SET-BY-CREATE
              public AdministrativeStateType   administrativeState;
                    // GET-REPLACE

**/

              public MONameSetType associatedImaLinkTPs;
                    // GET-REPLACE, ADD-REMOVE

              public EnumImaGroupState imaGroupNeState;
                    // GET
              public EnumImaGroupState imaGroupFeState;
                    // GET

              public EnumImaGroupSymmetry imaGroupSymmetry;
                    // GET, SET-BY-CREATE
              public long imaGroupMinNumTxLinks;
                    // GET-REPLACE
              public long imaGroupMinNumRxLinks;
                    // GET-REPLACE
              public long imaGroupNumTxCfgLinks;
                    // GET
              public long imaGroupNumRxCfgLinks;
                    // GET
              public long imaGroupNumTxActLinks;
                    // GET
              public long imaGroupNumRxActLinks;
                    // GET


              public EnumImaGroupTxClkMode imaGroupNeTxClkMode;
                    // GET-REPLACE
              public EnumImaGroupTxClkMode imaGroupFeTxClkMode;
                    // GET

              public MONameType imaGroupTxTimingRefLink;
                    // GET, SET-BY-CREATE
              public MONameType imaGroupRxTimingRefLink;
                    // GET

              public long imaGroupTxImaId;
```

```
                // GET, SET-BY-CREATE
            public long imaGroupRxImaId;
                // GET

            public EnumImaFrameLength imaGroupTxFrameLength;
                // GET-REPLACE
            public EnumImaFrameLength imaGroupRxFrameLength;
                // GET

            public long imaGroupDiffDelayMaxMS; // in milliseconds
                // GET-REPLACE

            public MONameType imaGroupLeastDelayLink;
                // Conditional
                // GET

            public long imaGroupDiffDelayMaxObs; // in milliseconds
                // GET-REPLACE (reset to zero)

            public long imaGroupAlphaValue;
                // GET-REPLACE
            public long imaGroupBetaValue;
                // GET-REPLACE
            public long imaGroupGammaValue;
                // GET-REPLACE

            public MONameType imaGroupTestLink;
                // conditional
                // present if IMA implements the Test Pattern Procedure.
                // GET-REPLACE
            public long imaGroupTestPattern;
                // conditional
                // present if IMA implements the Test Pattern Procedure.
                // GET-REPLACE
            public EnumImaTestProcStatus imaGroupTestProcStatus;
                // conditional
                // present if IMA implements the Test Pattern Procedure.
                // GET-REPLACE

            public long imaGroupTxAvailCellRate;
                // conditional
                // present, if implementation supports it
                // GET

            public long imaGroupRxAvailCellRate;
                // conditional
                // present, if implementation supports it
                // GET


        }; // valuetype AtmImaGroupTPValueType


        interface AtmImaGroupTP_F: itut_m3120::NetworkTP_F
        {
/**     AtmImaGroupTP_F inherits the following methods from
        itut_m3120:: TP_F:
        operationalStateGet, alarmStatusGet, containedInSubnetworkListGet,
        currentProblemListGet, alarmSeverityAssignmentProfilePointerGet,
        alarmSeverityAssignmentProfilePointerSet

        AtmImaGroupTP_F inherits the following methods from
        itut_m3120:: NetworkTP_F:
```

```
      pointDirectionalityGet, signalIdGet, administrativeStateGet,
      administrativeStateSet


**/


/**   Instances of AtmImaGroupTP_F are created using the AtmImaGroupTPFactory
      or automatically by the managed system.
**/

/**   The associated IMA Link TPs provides pointers to the associated IMA Link
      Termination Point instances belonging the this IMA Group
**/
            MONameSetType associatedImaLinkTPListGet
                  (in MONameType name)
                  raises (itut_x780::ApplicationError);
            void associatedImaLinkTPListSet
                  (in MONameType name,
                  in MONameSetType associatedImaLinkTPs)
                  raises (itut_x780::ApplicationError);
            void associatedImaLinkTPListAdd
                  (in MONameType name,
                  in MONameType newAssociatedImaLinkTP)
                  raises (itut_x780::ApplicationError);
            void associatedImaLinkTPListRemove
                  (in MONameType name,
                  in MONameType newAssociatedImaLinkTP)
                  raises (itut_x780::ApplicationError);

//    The current operational state of the near-end IMA Group State Machine
            EnumImaGroupState imaGroupNeStateGet
                  (in MONameType name)
                  raises (itut_x780::ApplicationError);

//    The current operational state of the far-end IMA Group State Machine
            EnumImaGroupState imaGroupFeStateGet
                  (in MONameType name)
                  raises (itut_x780::ApplicationError);

//    See "ATM Forum IMA v1.1, Section 5.2.2.7
            EnumImaGroupSymmetry imaGroupSymmetryGet
                  (in MONameType name)
                  raises (itut_x780::ApplicationError);

/**   Minimum number of transmit links required to be Active for the IMA
      group to be Operational.
**/
            long imaGroupMinNumTxLinksGet
                  (in MONameType name)
                  raises (itut_x780::ApplicationError);
            void imaGroupMinNumTxLinksSet
                  (in MONameType name,
                  in long imaGroupMinNumTxLinks)
                  raises (itut_x780::ApplicationError);

/**   Minimum number of receive links required to be Active for the IMA
      group to be Operational.
**/
            long imaGroupMinNumRxLinksGet
                  (in MONameType name)
                  raises (itut_x780::ApplicationError);
            void imaGroupMinNumRxLinksSet
```

```
                    (in MONameType name,
                    in long imaGroupMinNumRxLinks)
                    raises (itut_x780::ApplicationError);

/**    The number of links that are configured to transmit in this IMA
       group. This attribute overwrites the value of the
       imaGroupNumRxActLinks attribute when the IMA group is configured
       in the Symmetrical Configuration group symmetry mode.
**/
              long imaGroupNumTxCfgLinksGet
                    (in MONameType name)
                    raises (itut_x780::ApplicationError);

/**    The number of links that are configured to receive in this IMA
       group. This attribute is overwritten by the value of the
       imaGroupNumTxActLinks attribute when the IMA group is configured
       in the Symmetrical Configuration group symmetry mode.
**/
              long imaGroupNumRxCfgLinksGet
                    (in MONameType name)
                    raises (itut_x780::ApplicationError);

/**    The number of links which are configured to transmit and are
       currently Active in this IMA group.
**/
              long imaGroupNumTxActLinksGet
                    (in MONameType name)
                    raises (itut_x780::ApplicationError);

/**    The number of links which are configured to receive and are
       currently Active in this IMA group.
**/
              long imaGroupNumRxActLinksGet
                    (in MONameType name)
                    raises (itut_x780::ApplicationError);


//     Transmit clocking mode used by the near-end IMA group.
              EnumImaGroupTxClkMode imaGroupNeTxClkModeGet
                    (in MONameType name)
                    raises (itut_x780::ApplicationError);

              void imaGroupNeTxClkModeSet
                    (in MONameType name,
                    in EnumImaGroupTxClkMode imaGroupNeTxClkMode)
raises (itut_x780::ApplicationError);

//     Transmit clocking mode used by the far-end IMA group.
              EnumImaGroupTxClkMode imaGroupFeTxClkModeGet
                    (in MONameType name)
                    raises (itut_x780::ApplicationError);


/**    The transmit timing reference link to be used by the near-end for
       IMA data cell clock recovery from the ATM layer.
**/
              MONameType imaGroupTxTimingRefLinkGet
                    (in MONameType name)
                    raises (itut_x780::ApplicationError);

/**    The receive timing reference link to be used by the near-end for
       IMA data cell clock recovery from the ATM layer.
**/
```

```
              MONameType imaGroupRxTimingRefLinkGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);


//     The IMA ID currently in use by the near-end IMA function.
              long imaGroupTxImaIdGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);

//     The IMA ID currently in use by the far-end IMA function.
              long imaGroupRxImaIdGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);

/**    The frame length to be used by the IMA group in the transmit
       direction. Can only be set when the IMA group is startup.
**/
              EnumImaFrameLength imaGroupTxFrameLengthGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);

              void imaGroupTxFrameLengthSet
                     (in MONameType name,
                     in EnumImaFrameLength imaGroupTxFrameLength)
raises (itut_x780::ApplicationError);

/**    The frame length to be used by the IMA group in the receive
       direction.
**/
              EnumImaFrameLength imaGroupRxFrameLengthGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);

/**    The maximum number of milliseconds of differential delay among
       the links that will be tolerated on this interface.
**/
              long imaGroupDiffDelayMaxMSGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);

              long imaGroupDiffDelayMaxMSSet
                     (in MONameType name,
                     in long imaGroupDiffDelayMaxMS)
                     raises (itut_x780::ApplicationError);

/**    the link configured in the IMA group which has the smallest
       link propagation delay.
**/
              MONameType imaGroupLeastDelayLinkGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);

/**    The latest maximum differential delay observed (in milliseconds)
       between the links having the least and most link propagation
       delay, among the receive links that are currently configured in
       the IMA group.
**/
              long imaGroupDiffDelayMaxObsGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);

              void imaGroupDiffDelayMaxObsReset
```

```
                    (in MONameType name)
                    raises (itut_x780::ApplicationError);

/**    This indicates the 'alpha' value used to specify the number
       of consecutive invalid ICP cells to be detected before moving
       to the IMA Hunt state from the IMA Sync state.
**/
              long imaGroupAlphaValueGet
                    (in MONameType name)
                    raises (itut_x780::ApplicationError);

              void imaGroupAlphaValueSet
                    (in MONameType name,
                    in long imaGroupAlphaValue)
raises (itut_x780::ApplicationError);

/**    This indicates the 'beta' value used to specify the number
       of consecutive errored ICP cells to be detected before moving
       to the IMA Hunt state from the IMA Sync state.
**/
              long imaGroupBetaValueGet
                    (in MONameType name)
                    raises (itut_x780::ApplicationError);

              void imaGroupBetaValueSet
                    (in MONameType name,
                    in long imaGroupBetaValue)
raises (itut_x780::ApplicationError);

/**    This indicates the 'gamma' value used to specify the number
       of consecutive valid ICP cells to be detected before moving
       to the IMA Sync state from the IMA PreSync state.
**/
              long imaGroupGammaValueGet
                    (in MONameType name)
                    raises (itut_x780::ApplicationError);

              void imaGroupGammaValueSet
                    (in MONameType name,
                    in long imaGroupGammaValue)
raises (itut_x780::ApplicationError);

/**    This attribute is used to designate an interface as the test link
       for use in the Test Pattern Procedure.
**/           MONameType imaGroupTestLinkGet
                    (in MONameType name)
                    raises (itut_x780::ApplicationError);

              void imaGroupTestLinkSet
                    (in MONameType name,
                    in MONameType imaGroupTestLinkSet)
raises (itut_x780::ApplicationError);

/**    The value of this attribute is used to specify the Tx Test Pattern
       in an IMA group loopback operation. A value in the range 0 to
       255 designates a specific pattern. The distinguished value of
       -1 specifies that the implementation may choose the value.
**/
              long imaGroupTestPatternGet
                    (in MONameType name)
                    raises (itut_x780::ApplicationError);

              void imaGroupTestPatternSet
```

```
                    (in MONameType name,
                    in long imaGroupTestPattern)
                    raises (itut_x780::ApplicationError);

/**    This attribute is used to enable or disable the
       Test Pattern Procedure, and to note whether at least one
       link failed the test.
       The test is started by setting operating status. If any
       link should fail the test, the IMA will set the status to
       linkFail. The linkFail state will persist until
       either the disabled state is set or until no instance
       of imaLinkTestProcStatus has the value linkFail.
       Only the values disabled and operating may be written.
       Writing the operating value will not cause clearing of
       the linkFail state.
**/
              EnumImaTestProcStatus imaGroupTestProcStatusGet
                    (in MONameType name)
                    raises (itut_x780::ApplicationError);

              void imaGroupTestProcStatusSet
                    (in MONameType name,
                    in EnumImaTestProcStatus imaGroupTestProcStatus)
                    raises (itut_x780::ApplicationError);


/**    The current cell rate (truncated value in cells per second)
       provided by this IMA group in the transmit direction,
       considering all the transmit links in the Active state.
**/
              long imaGroupTxAvailCellRateGet
                    (in MONameType name)
                    raises (itut_x780::ApplicationError);

/**    The current cell rate (truncated value in cells per second)
       provided by this IMA group in the receive direction,
       considering all the receive links in the Active state.
**/
       long imaGroupRxAvailCellRateGet
                    (in MONameType name)
                    raises (itut_x780::ApplicationError);




              MANDATORY_NOTIFICATION(
                    itut_x780::Notifications, objectCreation)
              MANDATORY_NOTIFICATION(
                    itut_x780::Notifications, objectDeletion)
              MANDATORY_NOTIFICATION(
                    itut_x780::Notifications, attributeValueChange)
              MANDATORY_NOTIFICATION(
                    itut_x780::Notifications, stateChange)
              MANDATORY_NOTIFICATION(
                    itut_x780::Notifications, communicationAlarm)


       }; // interface AtmImaGroupTP_F


       interface AtmImaGroupTPFactory: itut_x780::ManagedObjectFactory
/*     EMS implementation may ignore a set of specific in parameters when
       creating an instance of AtmImaGroupTP, including:
```

```
       imaGroupSymmetry, imaGroupNeTxClkMode, imaGroupTxImaId,
       imaGroupTxFrameLength, imaGroupDiffDelayMaxMS, imaGroupAlphaValue,
       imaGroupBetaValue, and imaGroupGammaValue.
*/
       {
              itut_x780::ManagedObject create
                     (in NameBindingType nameBinding,
                            // module name containing Name Binding info.
                     in MONameType superior,
                            // Name of containing object.
                     in string reqID,
                            // Requested ID value for name, will be
                            // empty if auto-naming is to be used.
                     out MONameType name,
                            // Entire name of newly created object.
                     in StringSetType packageNameList,
                            // List of packages requested.

                     in AlarmSeverityAssignmentProfileNameType
                     alarmSeverityAssignmentProfilePointer,
                     in PointDirectionalityType pointDirectionality,
                     in SignalIdType signalId,
                     in AdministrativeStateType administrativeState,

                     in MONameSetType associatedImaLinkTPs,
                     in EnumImaGroupSymmetry imaGroupSymmetry,
                     in long imaGroupMinNumTxLinks,
                     in long imaGroupMinNumRxLinks,
                     in EnumImaGroupTxClkMode imaGroupNeTxClkMode,
                     in MONameType imaGroupTxTimingRefLink,
                     in long imaGroupTxImaId,
                     in EnumImaFrameLength imaGroupTxFrameLength,
                     in long imaGroupDiffDelayMaxMS,
                     in long imaGroupAlphaValue,
                     in long imaGroupBetaValue,
                     in long imaGroupGammaValue)

                     raises (itut_x780::ApplicationError,
                            itut_x780::CreateError);

       }; // interface AtmImaGroupTPFactory


/**
```

5.4.9 AtmImaLinkTP_F

```
*/
       valuetype AtmImaLinkTPValueType: itut_m3120::NetworkTPValueType
       {

       /**
       AtmImaLinkTPValueType uses the following inherited from
       itut_m3120:: TP:
              public MONameSetType             supportedByObjectList;
                     // GET
              public OperationalStateType             operationalState;
                     // conditional, present if an instance supports it.
                     // GET
              public AlarmStatusType                  alarmStatus;
                     // conditional, present if the TP supports communications
                     // alarm notification.
                     // GET
```

```
            public CurrentProblemSetType            currentProblemList;
                  // conditional, present if the TP supports communications
                  // alarm notification.
                  // GET
            public AlarmSeverityAssignmentProfileNameType
                  alarmSeverityAssignmentProfilePointer;
                  // conditional, present if an instance supports
                  // configuration of alarm severities.
                  // GET-REPLACE


      AtmImaLinkTPValueType uses the following inherited from
      itut_m3120:: NetworkTP:
            public PointDirectionalityType   pointDirectionality;
                  // GET
            public SignalIdType       signalId;
                  // GET
            public AdministrativeStateType   administrativeState;
                  // GET-REPLACE
            public Istring                   userLabel;
                  // conditional
                  // userLabelPackage
                  // GET-REPLACE


**/
            public MONameType imaLinkServerTTP;
                  // GET, SET-BY-CREATE

            public MONameType associatedImaGroupTP;
                  // GET-REPLACE

            public long imaLinkTxImaId;
                  // GET, SET-BY-CREATE
            public long imaLinkRxImaId;
                  // GET

            public EnumImaLinkState imaLinkNeTxState;
                  // GET
            public EnumImaLinkState imaLinkFeTxState;
                  // GET
            public EnumImaLinkState imaLinkNeRxState;
                  // GET
            public EnumImaLinkState imaLinkFeRxState;
                  // GET

            public long imaLinkTestPattern;
                  // conditional
                  // present if IMA implements the Test Pattern Procedure.
                  // GET-REPLACE
            public EnumImaTestProcStatus imaLinkTestProcStatus;
                  // conditional
                  // present if IMA implements the Test Pattern Procedure.
                  // GET-REPLACE


            public long imaLinkRelativeDelay; // in milliseconds
                  // conditional if implementation supports it.
                  // GET


      }; // valuetype AtmImaLinkTPValueType
```

```
      interface AtmImaLinkTP_F: itut_m3120::NetworkTP_F
      {
/**   AtmImaLinkTP_F inherits the following methods from
      itut_m3120:: TP_F:
      supportedByObjectListGet, operationalStateGet, alarmStatusGet,
      containedInSubnetworkListGet, currentProblemListGet,
      alarmSeverityAssignmentProfilePointerGet,
      alarmSeverityAssignmentProfilePointerSet

      AtmImaLinkTP_F inherits the following methods from
      itut_m3120:: NetworkTP_F:
      pointDirectionalityGet, signalIdGet, administrativeStateGet,
      administrativeStateSet, userLabelGet, userLabelSet


**/


/**   Instances of AtmImaLinkTP_F are created using the AtmImaLinkTPFactory
      or automatically by the managed system.
**/
            MONameType imaLinkServerTTPGet
                  (in MONameType name)
                  raises (itut_x780::ApplicationError);

            MONameType associatedImaGroupTPGet
                  (in MONameType name)
                  raises (itut_x780::ApplicationError);
            void associatedImaGroupTPSet
                  (in MONameType name,
                  in MONameType associatedImaGroupTP)
raises (itut_x780::ApplicationError);

            long imaLinkTxImaIdGet
                  (in MONameType name)
                  raises (itut_x780::ApplicationError);

            long imaLinkRxImaIdGet
                  (in MONameType name)
                  raises (itut_x780::ApplicationError);

            EnumImaLinkState imaLinkNeTxStateGet
                  (in MONameType name)
                  raises (itut_x780::ApplicationError);

            EnumImaLinkState imaLinkFeTxStateGet
                  (in MONameType name)
                  raises (itut_x780::ApplicationError);

            EnumImaLinkState imaLinkNeRxStateGet
                  (in MONameType name)
                  raises (itut_x780::ApplicationError);

            EnumImaLinkState imaLinkFeRxStateGet
                  (in MONameType name)
                  raises (itut_x780::ApplicationError);

            long imaLinkTestPatternGet
                  (in MONameType name)
                  raises (itut_x780::ApplicationError);

            void imaLinkTestPatternSet
```

```
                    (in MONameType name,
                    in long imaLinkTestPattern)
raises (itut_x780::ApplicationError);

              EnumImaTestProcStatus imaLinkTestProcStatusGet
                    (in MONameType name)
                    raises (itut_x780::ApplicationError);

              void imaLinkTestProcStatusSet
                    (in MONameType name,
                    in EnumImaTestProcStatus imaLinkTestProcStatus)
raises (itut_x780::ApplicationError);

              long imaLinkRelativeDelayGet
                    (in MONameType name)
                    raises (itut_x780::ApplicationError);



              MANDATORY_NOTIFICATION(
                    itut_x780::Notifications, objectCreation)
              MANDATORY_NOTIFICATION(
                    itut_x780::Notifications, objectDeletion)
              MANDATORY_NOTIFICATION(
                    itut_x780::Notifications, attributeValueChange)
              MANDATORY_NOTIFICATION(
                    itut_x780::Notifications, stateChange)
              MANDATORY_NOTIFICATION(
                    itut_x780::Notifications, communicationAlarm)


       }; // interface AtmImaLinkTP_F


       interface AtmImaLinkTPFactory: itut_x780::ManagedObjectFactory
       {
/*     EMS implementation may ignore a set of specific in parameters when
       creating an instance of AtmImaLinkTP, including:
       imaLinkTxImaId.
*/
              itut_x780::ManagedObject create
                    (in NameBindingType nameBinding,
                          // module name containing Name Binding info.
                    in MONameType superior,
                          // Name of containing object.
                    in string reqID,
                          // Requested ID value for name, will be
                          // empty if auto-naming is to be used.
                    out MONameType name,
                          // Entire name of newly created object.
                    in StringSetType packageNameList,
                          // List of packages requested.

                    in AlarmSeverityAssignmentProfileNameType
                    alarmSeverityAssignmentProfilePointer,
                    in PointDirectionalityType pointDirectionality,
                    in AdministrativeStateType administrativeState,
                    in Istring userLabel,

                    in MONameType imaLinkServerTTP,
                    in MONameType associatedImaGroupTP,
                    in long imaLinkTxImaId)
```

```
                raises (itut_x780::ApplicationError,
                       itut_x780::CreateError);

       }; // interface AtmImaLinkTPFactory


/**
```

5.4.10 AtmImaLinkTPPhy_F

```
*/
       valuetype AtmImaLinkTPPhyValueType: AtmImaLinkTPValueType
       {

       /**
       AtmImaLinkTPPhyValueType uses the following inherited from
       itut_m3120:: TP:
              public MONameSetType             supportedByObjectList;
                     // GET
              public OperationalStateType             operationalState;
                     // conditional, present if an instance supports it.
                     // GET
              public AlarmStatusType                  alarmStatus;
                     // conditional, present if the TP supports communications
                     // alarm notification.
                     // GET
              public CurrentProblemSetType            currentProblemList;
                     // conditional, present if the TP supports communications
                     // alarm notification.
                     // GET
              public AlarmSeverityAssignmentProfileNameType
                     alarmSeverityAssignmentProfilePointer;
                     // conditional, present if an instance supports
                     // configuration of alarm severities.
                     // GET-REPLACE


       AtmImaLinkTPPhyValueType uses the following inherited from
       itut_m3120:: NetworkTP:
              public PointDirectionalityType   pointDirectionality;
                     // GET
              public SignalIdType       signalId;
                     // GET
              public AdministrativeStateType   administrativeState;
                     // GET-REPLACE
              public Istring                   userLabel;
                     // conditional
                     // userLabelPackage
                     // GET-REPLACE


AtmImaLinkTPPhyValueType uses the following inherited from AtmImaLinkTPValueType

              public MONameType associatedImaGroupTP;
                     // GET-REPLACE

              public long imaLinkTxImaId;
                     // GET, SET-BY-CREATE
              public long imaLinkRxImaId;
                     // GET

              public EnumImaLinkState imaLinkNeTxState;
                     // GET
```

```
            public EnumImaLinkState imaLinkFeTxState;
                  // GET
            public EnumImaLinkState imaLinkNeRxState;
                  // GET
            public EnumImaLinkState imaLinkFeRxState;
                  // GET

            public long imaLinkTestPattern;
                  // conditional
                  // present if IMA implements the Test Pattern Procedure.
                  // GET-REPLACE
            public EnumImaTestProcStatus imaLinkTestProcStatus;
                  // conditional
                  // present if IMA implements the Test Pattern Procedure.
                  // GET-REPLACE


            public long imaLinkRelativeDelay; // in milliseconds
                  // conditional if implementation supports it.
                  // GET
**/

            public PortIDType leServerTTPPortID;
                  // conditional
                  // present if the server TTP port is represented
                  // GET-REPLACE
            public OperationalStateType leServerTTPOpState;
                  // conditional
                  // present if server TTP operational state is represented
                  // GET

            public PointCapacityType   potentialCapacity;
                  // conditional
                  // present if the TTP is a rate adaptive technology
                  // GET


      }; // valuetype AtmImaLinkTPPhyValueType


      interface AtmImaLinkTPPhy_F: itut_m3120::NetworkTP_F
      {
/**   AtmImaLinkTPPhy_F inherits the following methods from
      itut_m3120:: TP_F:
      supportedByObjectListGet, operationalStateGet, alarmStatusGet,
      containedInSubnetworkListGet, currentProblemListGet,
      alarmSeverityAssignmentProfilePointerGet,
      alarmSeverityAssignmentProfilePointerSet

      AtmImaLinkTPPhy_F inherits the following methods from
      itut_m3120:: NetworkTP_F:
      pointDirectionalityGet, signalIdGet, administrativeStateGet,
      administrativeStateSet, userLabelGet, userLabelSet

      AtmImaLinkTPPhy_F inherits the following methods from AtmImaLinkTP_F:
      AssociatedImaGroupTPGet, associatedImaGroupTPSet, imaLinkTxImaIdGet,
      imaLinkRxImaIdGet, imaLinkNeTxStateGet, imaLinkFeTxStateGet,
      imaLinkNeRxStateGet, imaLinkFeRxStateGet, imaLinkTestPatternGet,
      imaLinkTestPatternSet, imaLinkTestProcStatusGet, imaLinkTestProcStatusSet,
      imaLinkRelativeDelayGet

**/
```

```
/**    Instances of AtmImaLinkTPPhy_F are created using the AtmImaLinkTPPhyFactory
       or automatically by the managed system.
**/


              PortIDType serverTTPPortIDGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);

              void serverTTPPortIDSet
                     (in MONameType name,
                     in PortIDType serverTTPPortID )
                     raises (itut_x780::ApplicationError);

              OperationalStateType serverTTPOpStateGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);

/**
Provides potential bandwidth for rate adaptive server technology.
*/
              PointCapacityType potentialCapacityPackageGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);



              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, objectCreation)
              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, objectDeletion)
              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, attributeValueChange)
              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, stateChange)
              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, communicationAlarm)


       }; // interface AtmImaLinkTPPhy_F


       interface AtmImaLinkTPPhyFactory: itut_x780::ManagedObjectFactory
       {
/*     EMS implementation may ignore a set of specific in parameters when
       creating an instance of AtmImaLinkTPPhy, including:
       imaLinkTxImaId.
*/
              itut_x780::ManagedObject create
                     (in NameBindingType nameBinding,
                            // module name containing Name Binding info.
                     in MONameType superior,
                            // Name of containing object.
                     in string reqID,
                            // Requested ID value for name, will be
                            // empty if auto-naming is to be used.
                     out MONameType name,
                            // Entire name of newly created object.
                     in StringSetType packageNameList,
                            // List of packages requested.

                     in AlarmSeverityAssignmentProfileNameType
```

```
                    alarmSeverityAssignmentProfilePointer,
                    in PointDirectionalityType pointDirectionality,
                    in AdministrativeStateType administrativeState,
                    in Istring userLabel,

                    in MONameType associatedImaGroupTP,
                    in long imaLinkTxImaId,
                    in PortIDType leServerTTPPortID)

                    raises (itut_x780::ApplicationError,
                           itut_x780::CreateError);

      }; // interface AtmImaLinkTPPhyFactory


/**
```

5.4.11 AtmLink_F

```
*/

      valuetype AtmLinkValueType: itut_m3120::AbstractLinkValueType
      {
/**   AtmLinkValueType uses the following inherited from
      itut_m3120:: AbstractLinkValueType:
            public MONameType          aEnd;
                  // GET, SET-BY-CREATE
                  // points one of the atmLinkEnds terminating this link
            public CapacityType availableLinkCapacity;
                  // GET
            public SignalIdType signalId;
                  // GET, SET-BY-CREATE
            public MONameType          zEnd;
                  // points one of the atmLinkEnds terminating this link
                  // GET, SET-BY-CREATE
            public UsageCostType       usageCost;  // Link Weight
                  // conditional
                  // present if the link has an allocated usage cost
                  // GET-REPLACE
            public Istring             userLabel;
                  // conditional
                  // present  if a user label is supported
                  // GET-REPLACE
**/

/** AtmLinkValueType also supports the following:

            public AvailabilityStatusSetType availabilityStatus;
                  // GET
            public AdministrativeStateType adminState;
                  // GET-REPLACE
            public EnumRestorationModeType restorationMode;
                  // GET-REPLACE


      }; // valuetype AtmLinkValueType


      interface AtmLink_F: itut_m3120::AbstractLink_F
      {

/**   Instances of AtmLink are created using the AtmLinkFactory
      or automatically by the managed system.
```

```
**/

/**    AtmLink_F inherits the following methods from
       itut_m3120:: AbstractLink_F:
       aEndGet, availableLinkCapacityGet, signalIdGet, zEndGet, usageCostGet,
       usageCostSet, userLabelGet, userLabelSet

**/

              AvailabilityStatusSetType availabilityStatusGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);

              AdministrativeStateType administrativeStateGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);

              void administrativeStateSet
                     (in MONameType name,
                     in AdministrativeStateType adminstrativeState)
                     raises (itut_x780::ApplicationError);



/**
This read/write attribute is used to configure the restoration mode
of a link as: unavailable for routing and re-routing, available for
routing and not re-routing; available for re-routing and not
routing; or available for both routing and rerouting.
*/
              EnumRestorationModeType restorationModeGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);

              void restorationModeSet
                     (in MONameType name,
                     in EnumRestorationModeType enumRestorationMode)
                     raises (itut_x780::ApplicationError);




              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, objectCreation)
              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, objectDeletion)
              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, stateChange)
              CONDITIONAL_NOTIFICATION(
                     itut_x780::Notifications, attributeValueChange,
                     attributeValueChangeNotificationPackage)


       }; // interface AtmLink_F


       interface AtmLinkFactory: itut_x780::ManagedObjectFactory
       {
              itut_x780::ManagedObject create
                     (in NameBindingType nameBinding,
                            // module name containing Name Binding info.
                     in MONameType superior,
```

```
                        // Name of containing object.
                    in string reqID,
                        // Requested ID value for name, will be
                        // empty if auto-naming is to be used.
                    out MONameType name,
                        // Entire name of newly created object.
                    in StringSetType packageNameList,
                        // List of packages requested.


                    in MONameType aEnd,
                    in SignalIdType signalId,
                    in MONameType zEnd,
                    in UsageCostType usageCost,
                    in Istring userLabel,

                    in AdministrativeStateType adminState,
                    in EnumRestorationModeType restorationMode)

                    raises (itut_x780::ApplicationError,
                        itut_x780::CreateError);

        }; // interface AtmLinkFactory



/**
```

5.4.12 AtmLinkEnd_F

```
*/

        valuetype AtmLinkEndValueType: itut_m3120::AbstractLinkEndValueType
        {

/**     AtmLinkEndValueType uses the following inherited from
        itut_m3120:: AbstractLinkEndValueType:
                public PointCapacityType   availableLinkEndCapacity;
                        // GET
                        // Represents available ingress and egress bandwidth
                public AbstractLinkNameType             linkPointer;
                        // GET
                        // The linkPointer should be updated by the managed system
                        // whenever a link is created that is terminated by
                        // the AtmLinkEnd
                public SubnetworkNameSetType     containedInSubnetworkList;
                        // conditional
                        // present if this link end object instance is not
                        // named from a subnetwork managed object.
                        // GET-REPLACE, ADD-REMOVE
                public Istring                   userLabel;
                        // conditional
                        // present if a userLabel is supported.
                        // GET-REPLACE

                public PointCapacityType   potentialLinkEndCapacity;
                        // conditional
                        // present if LinkEnd is supported by an adaptive rate technology
                        // GET
```

```
*/
              public SignalIdType signalid;
                     // GET-REPLACE
              public AdministrativeStateType   administrativeState;
                     // GET-REPLACE
              public AvailabilityStatusSetType availabilityStatus;
                     // GET
              public AtmNetworkAccessProfileNameType networkAccessProfile;
                     // GET-REPLACE
              public EnumLinkEndType linkEndType;
                     // GET-REPLACE

              public long ingressMaxAssignableBW;
                     // GET
              public long egressMaxAssignableBW;
                     // GET

              public MONameSetType serverTTPList;
                     // GET, SET-BY-CREATE
              public AtmNetworkCTPNameSeqType supportedCTPs;
                     // GET

              public LoopbackLocationCodeSeqType loopbackLocID;
                     // conditional
                     // present if OAM Loopback is supported
                     // GET-REPLACE
              public VirtualIDType ilmiVpiVci;
                     // conditional
                     // present if ILMI is supported
                     // GET-REPLACE
              public long ilmiEstabConnectivityPollInterval;
                     // conditional
                     // present if ILMI is supported
                     // GET
              public long ilmiCheckConnectivityPollInterval;
                     // conditional
                     // present if ILMI is supported
                     // GET
              public long ilmiConnectivityPollFactor;
                     // conditional
                     // present if ILMI is supported
                     // GET
              public boolean cellScramblingEnabled;
                     // conditional
                     // present if TC Adaptor supports Cell Scrambling
                     // GET
              public MONameSetType vendorProfileList;
                     // conditional
                     // present if additional vendor configuration is supported
                     // GET-ADD-REMOVE


              public MONameSetType       logicalLinkEndList;
                     // conditional
                     // present if LinkEnd supports any logical link ends
                     // GET-ADD-REMOVE

              public MONameSetType       topologicalLinkEndList;
                     // conditional
                     // present if LinkEnd is supported by any topological link ends
                     // GET-ADD-REMOVE
```

```
      }; // valuetype AtmLinkEndValueType


      interface AtmLinkEnd_F: itut_m3120::AbstractLinkEnd
      {

/**   Instances of AtmLinkEnd are created using the AtmLinkEndFactory
      or automatically by the managed system.

      Instances of AtmLinkEnd may represent either Topological LinkEnds or Logical
      LinkEnds.  A Topological LinkEnd is the endpoint of a link that provides
      direct connectivity between subnetworks and is related one-to-one with an
      instance of a server TTP.  A Logical LinkEnd is derived from either many
      Topological LinkEnds, or a portion of a Topological LinkEnd.  The server
      TTP List of the Logical LinkEnd should point to the serverTTP of the
      Topological LinkEnds that support the Logical LinkEnd.
**/

/**   AtmLinkEnd _F inherits the following methods from
      itut_m3120:: AbstractLinkEnd_F:
      availableLinkEndCapacityGet, linkPointerGet, containedInSubnetworkListGet,
      containedInSubnetworkListSet, containedInSubnetworkListAdd,
      containedInSubnetworkListRemove, userLabelGet, userLabelSet,
      potentialLinkEndCapacityPackageGet

**/

            SignalIdType signalIdGet
                   (in MONameType name)
                  raises (itut_x780::ApplicationError);

            AvailabilityStatusSetType availabilityStatusGet
                   (in MONameType name)
                  raises (itut_x780::ApplicationError);

            AdministrativeStateType administrativeStateGet
                   (in MONameType name)
                  raises (itut_x780::ApplicationError);

            void administrativeStateSet
                  (in MONameType name,
                  in AdministrativeStateType adminstrativeState)
                  raises (itut_x780::ApplicationError);

            AtmNetworkAccessProfileNameType networkAccessProfileGet
                  (in MONameType name)
                  raises (itut_x780::ApplicationError);
            void networkAccessProfileSet
                  (in MONameType name,
                  in AtmNetworkAccessProfileNameType networkAccessProfile)
                  raises (itut_x780::ApplicationError);

            EnumLinkEndType linkEndTypeGet
                  (in MONameType name)
                  raises (itut_x780::ApplicationError);
            void linkEndTypeSet
                  (in MONameType name,
                  in EnumLinkEndType enumLinkEnd )
                  raises (itut_x780::ApplicationError);

            long ingressMaxAssignableBWGet
                  (in MONameType name)
                  raises (itut_x780::ApplicationError);
```

```
long egressMaxAssignableBWGet
       (in MONameType name)
       raises (itut_x780::ApplicationError);



MONameSetType serverTTPListGet
       (in MONameType name)
       raises (itut_x780::ApplicationError);

AtmNetworkCTPNameSeqType supportedCTPsGet
       (in MONameType name)
       raises (itut_x780::ApplicationError);

void supportedCTPAdd
       (in MONameType name,
       in AtmNetworkCTPNameType supportedCTP )
       raises (itut_x780::ApplicationError);

void supportedCTPRemove
       (in MONameType name,
       in AtmNetworkCTPNameType supportedCTP )
       raises (itut_x780::ApplicationError);

LoopbackLocationCodeSeqType loopbackLocIDGet
       (in MONameType name)
       raises (itut_x780::ApplicationError);
void loopbackLocIDSet
       (in MONameType name,
       in LoopbackLocationCodeSeqType loopbackloc)
       raises (itut_x780::ApplicationError);

VirtualIDType ilmiVpiVciGet
       (in MONameType name)
       raises (itut_x780::ApplicationError);
void ilmiVpiVciSet
       (in MONameType name,
       in VirtualIDType ilmiVpiVci)
       raises (itut_x780::ApplicationError);

long ilmiEstabConnectivityPollIntervalGet
       (in MONameType name)
       raises (itut_x780::ApplicationError);
void ilmiEstabConnectivityPollIntervalSet
       (in MONameType name,
       in long ilmiEstabInt)
       raises (itut_x780::ApplicationError);

long ilmiCheckConnectivityPollIntervalGet
       (in MONameType name)
       raises (itut_x780::ApplicationError);

void ilmiCheckConnectivityPollIntervalSet
       (in MONameType name,
       in long ilmiCheckInt)
       raises (itut_x780::ApplicationError);

long ilmiConnectivityPollFactorGet
       (in MONameType name)
       raises (itut_x780::ApplicationError);

void ilmiConnectivityPollFactorSet
```

```
                    (in MONameType name,
                    in long ilmiPollFactor)
                    raises (itut_x780::ApplicationError);

             boolean cellScramblingEnabledGet
                    (in MONameType name)
                    raises (itut_x780::ApplicationError);

             void cellScramblingEnabledSet
                    (in MONameType name,
                    in boolean cellScramblingEnabled)
                    raises (itut_x780::ApplicationError);


             MONameSetType vendorProfileListGet
                    (in MONameType name)
                    raises (itut_x780::ApplicationError);

             void vendorProfileListAdd
                    (in MONameType name,
                    in MONameSetType vendorProfileList )
                    raises (itut_x780::ApplicationError);

             void vendorProfileListRemove
                    (in MONameType name,
                    in MONameSetType vendorProfileList )
                    raises (itut_x780::ApplicationError);


/**
Provides pointers to the logical link end (if any) supported by a topologial link end
*/
             MONameSetType logicalLinkEndListGet
                    (in MONameType name)
                    raises (itut_x780::ApplicationError);

             void logicalLinkEndListSet
                    (in MONameType name,
                    in MONameSetType logicalLinkEndList)
                    raises (itut_x780::ApplicationError);

             void logicalLinkEndListAdd
                    (in MONameType name,
                    in MONameSetType logicalLinkEndList )
                    raises (itut_x780::ApplicationError);

             void logicalLinkEndListRemove
                    (in MONameType name,
                    in MONameSetType logicalLinkEndList )
                    raises (itut_x780::ApplicationError);

/**
Provides pointers to the topological link end supporting the logical link end
*/
             MONameSetType topologicalLinkEndListGet
                    (in MONameType name)
                    raises (itut_x780::ApplicationError);

             void topologicalLinkEndListSet
                    (in MONameType name,
                    in MONameSetType topologicalLinkEndList)
                    raises (itut_x780::ApplicationError);
```

```
        void topologicalLinkEndListAdd
              (in MONameType name,
              in MONameSetType topologicalLinkEndList )
              raises (itut_x780::ApplicationError);

        void topologicalLinkEndListRemove
              (in MONameType name,
              in MONameSetType topologicalLinkEndList )
              raises (itut_x780::ApplicationError);

        void linkPVCTrace
              (in MONameType name,
              in EnumLinkTraceType enumLinkTrace,
              in AtmSubnetworkNameSeqType selectedSubnets,
              out AtmSNCNameSeqType traceResults)
              raises (itut_x780::ApplicationError);

        MANDATORY_NOTIFICATION(
              itut_x780::Notifications, objectCreation)
        MANDATORY_NOTIFICATION(
              itut_x780::Notifications, objectDeletion)
        MANDATORY_NOTIFICATION(
              itut_x780::Notifications, attributeValueChange)
        MANDATORY_NOTIFICATION(
              itut_x780::Notifications, stateChange)

    }; // interface AtmLinkEnd_F


    interface AtmLinkEndFactory: itut_x780::ManagedObjectFactory
    {
        itut_x780::ManagedObject create
              (in NameBindingType nameBinding,
                    // module name containing Name Binding info.
              in MONameType superior,
                    // Name of containing object.
              in string reqID,
                    // Requested ID value for name, will be
                    // empty if auto-naming is to be used.
              out MONameType name,
                    // Entire name of newly created object.
              in StringSetType packageNameList,
                    // List of packages requested.

              in PointCapacityType       availableLinkEndCapacity,
              in AbstractLinkNameType           linkPointer,
              in SubnetworkNameSetType   containedInSubnetworkList,
              in Istring                 userLabel,


              in SignalIdType signalId,
              in AdministrativeStateType administrativeState,

              in AtmNetworkAccessProfileNameType networkAccessProfile,
              in EnumLinkEndType leType,

              in MONameType serverTTP,
              in AtmNetworkCTPNameSeqType supportedCTPs,
              in LoopbackLocationCodeSeqType loopbackLocID,
                    // conditional

              in VirtualIDType leIlmiVpiVci,
                    // conditional
```

```
                    in     long ilmiEstabConnectivityPollInterval,
                           // conditional
                    in     long ilmiConnectivityPollFactor,
                           // conditional

                    in MONameSetType vendorProfile)
                           // conditional
                    raises (itut_x780::ApplicationError,
                           itut_x780::CreateError);

      }; // interface AtmLinkEndFactory



/**
```

5.4.13 AtmLinkEndPhy_F

```
*/
      valuetype AtmLinkEndPhyValueType: AtmLinkEndValueType
      {

            public CharacteristicInfoType leServerTTPCharInfo;
                  // GET, SET-BY-CREATE
            public PortIDType leServerTTPPortID;
                  // conditional
                  // present if the server TTP port is represented
                  // GET-REPLACE
            public OperationalStateType leServerTTPOpState;
                  // conditional
                  // present if operational state of the server TTP is represented
                  // GET

            public AlarmSeverityAssignmentProfileNameType profile;
                  // conditional
                  // GET-REPLACE

            public CurrentProblemSetType currentProblemList;
                  // conditional
                  // GET-REPLACE, ADD-REMOVE

      }; // valuetype AtmLinkEndPhyValueType


      interface AtmLinkEndPhy_F:AtmLinkEnd_F
      {
/**   Instances of AtmLinkEndPhy are created using the AtmLinkEndPhyFactory
      or automatically by the managed system.
**/

            CharacteristicInfoType serverTTPCharInfoGet
                  (in MONameType name)
                  raises (itut_x780::ApplicationError);

            PortIDType serverTTPPortIDGet
                  (in MONameType name)
                  raises (itut_x780::ApplicationError);

            void serverTTPPortIDSet
                  (in MONameType name,
                  in PortIDType serverTTPPortID )
                  raises (itut_x780::ApplicationError);
```

```
        OperationalStateType serverTTPOpStateGet
               (in MONameType name)
               raises (itut_x780::ApplicationError);


        AlarmSeverityAssignmentProfileNameType profileGet
               (in MONameType name)
               raises (itut_x780::ApplicationError);

        AlarmSeverityAssignmentProfileNameType profileSet
               (in MONameType name,
               in AlarmSeverityAssignmentProfileNameType asapName)
               raises (itut_x780::ApplicationError);

        CurrentProblemSetType currentProblemListGet
               (in MONameType name)
               raises (itut_x780::ApplicationError);

        MANDATORY_NOTIFICATION(
               itut_x780::Notifications, objectCreation)
        MANDATORY_NOTIFICATION(
               itut_x780::Notifications, objectDeletion)
        MANDATORY_NOTIFICATION(
               itut_x780::Notifications, attributeValueChange)
        MANDATORY_NOTIFICATION(
               itut_x780::Notifications, stateChange)
        CONDITIONAL_NOTIFICATION(
               itut_x780::Notifications, communicationAlarm,
               tmnCommunicationsAlarmInformationR1Package)



};  // interface AtmLinkEndPhy


interface AtmLinkEndPhyFactory: itut_x780::ManagedObjectFactory
{
        itut_x780::ManagedObject create
               (in NameBindingType nameBinding,
                      // module name containing Name Binding info.
               in MONameType superior,
                      // Name of containing object.
               in string reqID,
                      // Requested ID value for name, will be
                      // empty if auto-naming is to be used.
               out MONameType name,
                      // Entire name of newly created object.
               in StringSetType packageNameList,
                      // List of packages requested.



               in PointCapacityType       availableLinkEndCapacity,
               in AbstractLinkNameType           linkPointer,
               in SubnetworkNameSetType   containedInSubnetworkList,
               in Istring                 userLabel,



               in SignalIdType signalId,
               in AdministrativeStateType administrativeState,
```

```
                    in AtmNetworkAccessProfileNameType networkAccessProfile,
                    in EnumLinkEndType leType,

                    in MONameType leServerTTP,
                    in AtmNetworkCTPNameSeqType supportedCTPs,
                    in LoopbackLocationCodeSeqType loopbackLocID,
                           // conditional

                    in VirtualIDType ilmiVpiVci,
                           // conditional
                    in     long ilmiEstabConnectivityPollInterval,
                           // conditional
                    in     long ilmiConnectivityPollFactor,
                           // conditional

                    in MONameType vendorProfile,

                    in CharacteristicInfoType serverTTPCharInfo,
                    in PortIDType serverTTPPortID,
                    in AlarmSeverityAssignmentProfileNameType profile)

                    raises (itut_x780::ApplicationError,
                           itut_x780::CreateError);

      }; // interface AtmLinkEndPhyFactory




/**
```

5.4.14 AtmNetworkAccessProfile_F

```
*/
      valuetype AtmNetworkAccessProfileValueType:
             itut_x780::ManagedObjectValueType
      {

             public long totalIngressBW;
                    // GET-REPLACE
             public long totalEgressBW;
                    // GET-REPLACE
             public long maxNumActiveVcConn;
                    // GET-REPLACE
             public long maxNumActiveVpConn;
                    // GET-REPLACE
             public VpiOrVciRangeType vpiRange;
                    // conditional, present if VPI range partitioning is supported
                    // GET-REPLACE
             public VpiOrVciRangeType vciRange;
                    // conditional, present if VCI range partitioning is supported
                    // GET-REPLACE

      }; // valuetype AtmNetworkAccessProfileValueType


      interface AtmNetworkAccessProfile_F: itut_x780::ManagedObject_F
      {
/**
      Instances of AtmNetworkAccessProfile are created using the
```

```
       AtmNetworkAccessProfileFactory or by the managed system.


An atmNetworkAccessProfile contains information that describe the maximum
ingress and egress bandwidth, along with the range of VPI or VCI values that
are applied to the atmLink or atmLinkEnd object instances that point to it.
**/

/**
This read/write attribute identifies the total aggregate ingress
bandwidth for a link or a  LinkEnd.
*/
              long totalIngressBWGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);

/**
This read/write attribute identifies the total aggregate egress
bandwidth for a link or a LinkEnd.
*/
              long totalEgressBWGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);


/**
This read/write attribute identifies the maximum number of
concurrently active VC connections that a link or a LinkEnd may support.
*/
              long maxNumActiveVcConnGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);


/**
This read/write attribute identifies the maximum number of
concurrently active VP connections that a link or a LinkEnd may support.
*/
              long maxNumActiveVpConnGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);


/**
This read/write attribute describes the virtual ID range (VPIs) that
may be used for Connections associated with a link or LinkEnd.
*/
              VpiOrVciRangeType vpiRangeGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);


/**
This read/write attribute describes the virtual ID range (VCIs) that
may be used for Connections associated with a link or LinkEnd.
*/
              VpiOrVciRangeType vciRangeGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);


              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, objectCreation)
```

```
            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectDeletion)


      }; // interface AtmNetworkAccessProfile_F


      interface AtmNetworkAccessProfileFactory:
            itut_x780::ManagedObjectFactory
      {
            itut_x780::ManagedObject create
                  (in NameBindingType nameBinding,
                        // module name containing Name Binding info.
                  in MONameType superior,
                        // Name of containing object.
                  in string reqID,
                        // Requested ID value for name, will be
                        // empty if auto-naming is to be used.
                  out MONameType name,
                        // Entire name of newly created object.
                  in StringSetType packageNameList,
                        // List of packages requested.

                  in long totalIngressBW,
                  in long totalEgressBW,
                  in long maxNumActiveVcConn,
                  in long maxNumActiveVpConn,
                  in VpiOrVciRangeType vpiRange,
                  in VpiOrVciRangeType vciRange )
                  raises (itut_x780::ApplicationError,
                        itut_x780::CreateError);

      }; // interface AtmNetworkAccessProfileFactory



/**
```

5.4.15 AtmNetworkTP_F

```
*/
      valuetype AtmNetworkTPValueType:
            itut_m3120::NetworkTPValueType

      {
/**   AtmNetworkTPValueType uses the following inherited from
      itut_m3120:: TP:
            public OperationalStateType            operationalState;
                  // conditional, present if an instance supports it.
                  // GET
            public AlarmStatusType                 alarmStatus;
                  // conditional, present if the TP supports communications
                  // alarm notification.
                  // GET
            public CurrentProblemSetType           currentProblemList;
                  // conditional, present if the TP supports communications
                  // alarm notification.
                  // GET
            public AlarmSeverityAssignmentProfileNameType
                  alarmSeverityAssignmentProfilePointer;
                  // conditional, present if an instance supports
                  // configuration of alarm severities.
```

```
                    // GET-REPLACE


       AtmNetworkTPValueType uses the following inherited from
       itut_m3120:: NetworkTP:
              public PointDirectionalityType   pointDirectionality;
                    // GET
              public SignalIdType        signalId;
                    // GET, SET-BY-CREATE
              public SNCNameSetType            sncPointer;
                    // GET


**/

              public VirtualIDType networkTPVpiVci;
                    // GET
              public EnumPmOamMethodType pmOamMethod;
                    // GET-REPLACE
                    // Conditional, present if PM-OAM supported
              public EnumPmOamDirectionType pmOamDirection;
                    // GET-REPLACE
                    // Conditional, present if PM-OAM supported
              public EnumPmOamBlockSizeType pmOamBlockSize;
                    // GET-REPLACE
                    // Conditional, present if PM-OAM supported
              public BooleanTypeOpt pmOamForwardActive;
                    // GET-REPLACE
                    // Conditional, present if PM-OAM supported
              public BooleanTypeOpt pmOamBackwardActive;
                    // GET-REPLACE
                    // Conditional, present if PM-OAM supported

       }; // valuetype AtmNetworkTPValueType


       interface AtmNetworkTP_F: itut_m3120::NetworkTP_F
       {

/**    AtmNetworkTP_F inherits the following methods from
       itut_m3120:: TP_F:
       operationalStateGet, alarmStatusGet, containedInSubnetworkListGet,
       currentProblemListGet, alarmSeverityAssignmentProfilePointerGet,
       alarmSeverityAssignmentProfilePointerSet

       AtmNetworkTP_F inherits the following methods from
       itut_m3120:: NetworkTP_F:
       pointDirectionalityGet, signalIdGet, sncPointerGet


**/

/**
This read-only attribute identifies the VPI/VCI value associated
with the connection being terminated
*/
              VirtualIDType networkTPVpiVciGet
                    (in MONameType name)
                    raises (itut_x780::ApplicationError);

/**
This operation is used to request that the NetworkTP insert a
loopback OAM cell into the ATM cell stream, verify its return, and
```

```
report the results of the loopback (i.e., passed or failed) back to
the management system. Along with each request will be the location
where the inserted OAM cell shall loop-back and an indication as to
whether a segment or end-to-end OAM cell shall be used.  The
Loopback Location Code which indicates where the loopback is to take
place may be used to identify the loopback location.  Additionally,
a globally unique default value (e.g., "end-point") may also be used
to perform a loopback at the other end of a VCC or VPC.
*/

              LoopbackCellReplyType loopbackOamCell
                     (in MONameType name,
                     in LoopbackLocType loopbackLoc,
                     in EnumOamLoopbackDirectionType loopbackDirection,
                     in EnumOamCellType oamCellType )
                     raises (itut_x780::ApplicationError);


/**
This optional attribute indicates the method used to setup and
terminate the PM OAM monitoring activity.  Valid values are TMN,
OAM, or notSupported.   If the value is notSupported, then PM OAM is
not supported on the endpoint
*/
              EnumPmOamMethodType pmOamMethodGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);

              void pmOamMethodSet
                     (in MONameType name,
                     in EnumPmOamMethodType enumPmOamMethod )
                     raises (itut_x780::ApplicationError);

/**
This optional attribute indicates the desired direction(s) of
transmission to monitor PM OAM.  Valid directions are:  away from
activator (transmit), towards activator (receive), or both
*/
              EnumPmOamDirectionType pmOamDirectionGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);

              void pmOamDirectionSet
                     (in MONameType name,
                     in EnumPmOamDirectionType enumPmOamDirection )
                     raises (itut_x780::ApplicationError);

/**
This optional attribute indicates the PM OAM nominal block size
choice for both the receive and transmit dirctions
*/
              EnumPmOamBlockSizeType pmOamBlockSizeGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);

              void pmOamBlockSizeSet
                     (in MONameType name,
                     in EnumPmOamBlockSizeType pmBlockSize)
                     raises (itut_x780::ApplicationError);

/**
This optional boolean attribute is used to initiate generation of PM
OAM cells in the forward direction by setting the value to true
```

```
*/
              boolean pmOamForwardActiveGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);

              void pmOamForwardActiveSet
                     (in MONameType name,
                     in boolean pmForwardActive )
                     raises (itut_x780::ApplicationError);

/**
This optional boolean attribute is used to initiate generation of PM
OAM cells in the backward direction by setting the value to true
*/
              boolean pmOamBackwardActiveGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);

              void pmOamBackwardActiveSet
                     (in MONameType name,
                     in boolean pmBackwardActive )
                     raises (itut_x780::ApplicationError);

              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, objectCreation)
              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, objectDeletion)
              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, attributeValueChange)
              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, stateChange)


       }; // interface AtmNetworkTP_F


/**
```

5.4.16 AtmNetworkCTP_F

```
*/
       valuetype AtmNetworkCTPValueType:AtmNetworkTPValueType

       {
/**    AtmNetworkCTPValueType uses the following inherited from
       itut_m3120:: TPValueType:
              public OperationalStateType             operationalState;
                     // conditional, present if an instance supports it.
                     // GET
              public AlarmStatusType                  alarmStatus;
                     // conditional, present if the TP supports communications
                     // alarm notification.
                     // GET
              public CurrentProblemSetType            currentProblemList;
                     // conditional, present if the TP supports communications
                     // alarm notification.
                     // GET
              public AlarmSeverityAssignmentProfileNameType
                     alarmSeverityAssignmentProfilePointer;
                     // conditional, present if an instance supports
                     // configuration of alarm severities.
                     // GET-REPLACE
```

```
      AtmNetworkCTPValueType uses the following inherited from
      itut_m3120:: NetworkTPValueType:
            public PointDirectionalityType   pointDirectionality;
                  // GET
            public SignalIdType        signalId;
                  // GET, SET-BY-CREATE
            public SNCNameSetType            sncPointer;
                  // GET

      AtmNetworkCTPValueType uses the following inherited from AtmNetworkTPValueType:
            public VirtualIDType networkTPVpiVci;
                  // GET
            public EnumPmOamMethodType pmOamMethod;
                  // GET-REPLACE
                  // Conditional, present if PM-OAM supported
            public EnumPmOamDirectionType pmOamDirection;
                  // GET-REPLACE
                  // Conditional, present if PM-OAM supported
            public EnumPmOamBlockSizeType pmOamBlockSize;
                  // GET-REPLACE
                  // Conditional, present if PM-OAM supported
            public BooleanTypeOpt pmOamForwardActive;
                  // GET-REPLACE
                  // Conditional, present if PM-OAM supported
            public BooleanTypeOpt pmOamBackwardActive;
                  // GET-REPLACE
                  // Conditional, present if PM-OAM supported

**/




            public AtmAbstractTrafficDescNameType egressProfile;
                  // GET-REPLACE
            public AtmAbstractTrafficDescNameType ingressProfile;
                  // GET-REPLACE
            public boolean segmentEndpoint;
                  // GET-REPLACE
            public AtmNetworkTTPNameType relatedAtmTTP;
                  // GET-REPLACE
            public BooleanTypeOpt ingressTaggingInd;
                  // GET-REPLACE
            public BooleanTypeOpt egressTaggingInd;
                  // GET-REPLACE



      }; // valuetype AtmNetworkCTPValueType


      interface AtmNetworkCTP_F: AtmNetworkTP_F
      {
/**   Instances of AtmNetworkCTP are created by the managed system as a
      result of setting up a SNC, or by using the Factory.
**/

/**   AtmNetworkCTP_F inherits the following methods from
      itut_m3120:: TP_F:
```

```
       operationalStateGet, alarmStatusGet, containedInSubnetworkListGet,
       currentProblemListGet, alarmSeverityAssignmentProfilePointerGet,
       alarmSeverityAssignmentProfilePointerSet

       AtmNetworkCTP_F inherits the following methods from
       itut_m3120:: NetworkTP_F:
       pointDirectionalityGet, signalIdGet, sncPointerGet

       AtmNetworkCTP_F inherits the following methods from AtmNetworkTP_F:
       loopbackOamCell, networkTPVpiVciGet, pmOamMethodGet, pmOamMethodSet,
       pmOamDirectionGet, pmOamDirectionSet, pmOamBlockSizeGet, pmOamBlockSizeSet,
       pmOamForwardActiveGet, pmOamForwardActiveSet, pmOamBackwardActiveGet,
       pmOamBackwardActiveSet


**/


/**
This boolean attribute indicates whether the NetworkCTP object
instance has been configured to represent an end-point of a VCC or
VPC Segment
*/
             boolean segmentEndpointGet
                    (in MONameType name)
                    raises (itut_x780::ApplicationError);
             void segmentEndpointSet
                    (in MONameType name,
                    in boolean segmentEndpoint)
                    raises (itut_x780::ApplicationError);

             AtmNetworkTTPNameType relatedAtmTTPGet
                    (in MONameType name)
                    raises (itut_x780::ApplicationError);
             void relatedAtmTTPSet
                    (in MONameType name,
                    in AtmNetworkTTPNameType relatedAtmTTP)
                    raises (itut_x780::ApplicationError);


/**
one instance of the Traffic Descriptor managed entity may
characterize the CTP.
*/
             AtmAbstractTrafficDescNameType egressTrafficDescProfileGet
                    (in MONameType name)
                    raises (itut_x780::ApplicationError);

             void egressTrafficDescProfileSet
                    (in MONameType name,
                    in AtmAbstractTrafficDescNameType egressProfile)
                    raises (itut_x780::ApplicationError);

             AtmAbstractTrafficDescNameType ingressTrafficDescProfileGet
                    (in MONameType name)
                    raises (itut_x780::ApplicationError);

             void ingressTrafficDescProfileSet
                    (in MONameType name,
                    in AtmAbstractTrafficDescNameType ingressProfile)
                    raises (itut_x780::ApplicationError);
```

```
/**
This boolean attribute specifies if tagging is being used
on the receive side of an ATM VPC or VCC.  A value of true indicates
that tagging is being used.  A value of false indicates it is not being used.
*/
              boolean ingressTaggingIndGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);

              void ingressTaggingIndSet
                     (in MONameType name,
                     in boolean ingressTagInd)
                     raises (itut_x780::ApplicationError);

/**
This boolean attribute specifies if tagging is being used
on the transmit side of an ATM VPC or VCC.  A value of true
indicates that tagging is being used.  A value of false indicates it
is not being used.
*/
              boolean egressTaggingIndGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);

              void egressTaggingIndSet
                     (in MONameType name,
                     in boolean egressTagInd)
                     raises (itut_x780::ApplicationError);


              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, objectCreation)
              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, objectDeletion)
              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, attributeValueChange)
              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, stateChange)
              CONDITIONAL_NOTIFICATION(
                     itut_x780::Notifications, communicationAlarm,
                     tmnCommunicationsAlarmInformationR1Package)


       }; // interface AtmNetworkCTP_F


       interface AtmNetworkCTPFactory:
              itut_x780::ManagedObjectFactory
       {
              itut_x780::ManagedObject create
                     (in NameBindingType nameBinding,
                            // module name containing Name Binding info.
                     in MONameType superior,
                            // Name of containing object.
                     in string reqID,
                            // Requested ID value for name, will be
                            // empty if auto-naming is to be used.
                     out MONameType name,
                            // Entire name of newly created object.
                     in StringSetType packageNameList,
                            // List of packages requested.

                     in AlarmSeverityAssignmentProfileNameType
```

```
                    alarmSeverityAssignmentProfilePointer,
                    in PointDirectionalityType pointDirectionality,
                    in SignalIdType signalId,

                    in VirtualIDType networkTPVpiVci,
                    in EnumPmOamMethodType pmOamMethod,
                    in EnumPmOamDirectionType pmOamDirection,
                    in EnumPmOamBlockSizeType pmOamBlockSize,
                    in BooleanTypeOpt pmOamForwardActive,
                    in BooleanTypeOpt pmOamBackwardActive,

                    in AtmAbstractTrafficDescNameType egressProfile,
                    in AtmAbstractTrafficDescNameType ingressProfile,
                    in boolean segmentEndpoint,
                    in AtmNetworkTTPNameType relatedAtmTTP,
                    in BooleanTypeOpt ingressTaggingInd,
                    in BooleanTypeOpt egressTaggingInd)

                    raises (itut_x780::ApplicationError,
                           itut_x780::CreateError);

      }; // interface AtmNetworkCTPFactory




/**
```

5.4.17 AtmNetworkTTP_F

```
*/

      valuetype AtmNetworkTTPValueType: AtmNetworkTPValueType
      {
/**   AtmNetworkTTPValueType uses the following inherited from
      itut_m3120:: TPValueType:
            public OperationalStateType             operationalState;
                  // conditional, present if an instance supports it.
                  // GET
            public AlarmStatusType                  alarmStatus;
                  // conditional, present if the TP supports communications
                  // alarm notification.
                  // GET
            public CurrentProblemSetType            currentProblemList;
                  // conditional, present if the TP supports communications
                  // alarm notification.
                  // GET
            public AlarmSeverityAssignmentProfileNameType
                  alarmSeverityAssignmentProfilePointer;
                  // conditional, present if an instance supports
                  // configuration of alarm severities.
                  // GET-REPLACE


      AtmNetworkTTPValueType uses the following inherited from
      itut_m3120:: NetworkTPValueType:
            public PointDirectionalityType   pointDirectionality;
                  // GET
            public SignalIdType       signalId;
                  // GET, SET-BY-CREATE
            public SNCNameSetType            sncPointer;
                  // GET
```

```
      AtmNetworkTTPValueType uses the following inherited from AtmNetworkTPValueType:
            public VirtualIDType networkTPVpiVci;
                  // GET
            public EnumPmOamMethodType pmOamMethod;
                  // GET-REPLACE
                  // Conditional, present if PM-OAM supported
            public EnumPmOamDirectionType pmOamDirection;
                  // GET-REPLACE
                  // Conditional, present if PM-OAM supported
            public EnumPmOamBlockSizeType pmOamBlockSize;
                  // GET-REPLACE
                  // Conditional, present if PM-OAM supported
            public BooleanTypeOpt pmOamForwardActive;
                  // GET-REPLACE
                  // Conditional, present if PM-OAM supported
            public BooleanTypeOpt pmOamBackwardActive;
                  // GET-REPLACE
                  // Conditional, present if PM-OAM supported

**/
            public AtmNetworkCTPNameType relatedAtmCTP;
                  // GET

            public MONameSetType clientLinkEndPointerList;
                  // GET-REPLACE
            public MONameType   serviceProfilePointer;
                  // GET-REPLACE
            public MONameType   aalProfilePointer;
                  // GET-REPLACE


      }; // valuetype AtmNetworkTTPValueType


      interface AtmNetworkTTP_F: AtmNetworkTP_F
      {
/**   Instances of AtmNetworkTTP are created by the managed system as a
      result of setting up a SNC with trail endpoints, or by using the
      factory.  An ATM SNC terminated by two CTPs each with associated
      TTPs represents an end-to-end ATM Trail.
**/
/**   AtmNetworkTTP_F inherits the following methods from
      itut_m3120:: TP_F:
      operationalStateGet, alarmStatusGet, containedInSubnetworkListGet,
      currentProblemListGet, alarmSeverityAssignmentProfilePointerGet,
      alarmSeverityAssignmentProfilePointerSet

      AtmNetworkTTP_F inherits the following methods from
      itut_m3120:: NetworkTP_F:
      pointDirectionalityGet, signalIdGet, sncPointerGet

      AtmNetworkTTP_F inherits the following methods from AtmNetworkTP_F:
      LoopbackOamCell, networkTPVpiVciGet, pmOamMethodGet, pmOamMethodSet,
      pmOamDirectionGet, pmOamDirectionSet, pmOamBlockSizeGet, pmOamBlockSizeSet,
      pmOamForwardActiveGet, pmOamForwardActiveSet, pmOamBackwardActiveGet,
      pmOamBackwardActiveSet

      This managed object shall send a communicationsAlarm notification to the
      managing system, when the cellLossIntegrationPeriod (identified by aalProfile)
      expires.
```

```
**/

              AtmNetworkCTPNameType relatedAtmCTPGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);

              MONameSetType clientLinkEndPointerListGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);
              void clientLinkEndPointerListSet
                     (in MONameType name,
                     in MONameSetType clientLinkEndPointerList)
                     raises (itut_x780::ApplicationError);

/**
This attribute provides a pointer to a managed object instance that provides
information used to control service interworking (e.g., a cesServiceProfile object).
*/
              MONameType serviceProfilePointerGet(in MONameType name)
                     raises (itut_x780::ApplicationError);
              void serviceProfilePointerSet(in MONameType name,
                     in MONameType serviceProfilePointer)
                     raises (itut_x780::ApplicationError);

/**
This attribute provides a pointer to an aalProfile managed object instance that
defines the common ATM Adaptation Layer processing needed.
*/

              MONameType aalProfilePointerGet(in MONameType name)
                     raises (itut_x780::ApplicationError);
              void AalProfilePointerSet(in MONameType name,
                     in MONameType aalProfilePointer)
                     raises (itut_x780::ApplicationError);


              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, objectCreation)
              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, objectDeletion)
              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, attributeValueChange)
              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, stateChange)
              CONDITIONAL_NOTIFICATION(
                     itut_x780::Notifications, communicationAlarm,
                     tmnCommunicationsAlarmInformationR1Package)



       }; // interface AtmNetworkTTP_F



       interface AtmNetworkTTPFactory:
              itut_x780::ManagedObjectFactory
       {
              itut_x780::ManagedObject create
                     (in NameBindingType nameBinding,
                            // module name containing Name Binding info.
                     in MONameType superior,
                            // Name of containing object.
```

```
                    in string reqID,
                          // Requested ID value for name, will be
                          // empty if auto-naming is to be used.
                    out MONameType name,
                          // Entire name of newly created object.
                    in StringSetType packageNameList,
                          // List of packages requested.

                    in AlarmSeverityAssignmentProfileNameType
                    alarmSeverityAssignmentProfilePointer,
                    in PointDirectionalityType pointDirectionality,
                    in SignalIdType signalId,

                    in VirtualIDType networkTPVpiVci,
                    in EnumPmOamMethodType pmOamMethod,
                    in EnumPmOamDirectionType pmOamDirection,
                    in EnumPmOamBlockSizeType pmOamBlockSize,
                    in BooleanTypeOpt pmOamForwardActive,
                    in BooleanTypeOpt pmOamBackwardActive,
                    in AtmNetworkCTPNameType relatedAtmCTP,
                    in MONameSetType clientLinkEndPointerList,
                    in MONameTypeserviceProfilePointer,
                    in AalProfileNameType aalProfilePointer)

                    raises (itut_x780::ApplicationError,
                          itut_x780::CreateError);

      }; // interface AtmNetworkTTPFactory


/**
```

5.4.18 AtmRoutingProfile_F

```
*/
      valuetype AtmRoutingProfileValueType: itut_x780::ManagedObjectValueType
      {
            public EnumConnectionType connectionType;
                  // GET, SET-BY-CREATE
            public RouteDescriptionListType routeDescriptionList;
                  // GET, SET-BY-CREATE
            public unsigned short maxHops;
                  // GET, SET-BY-CREATE

      }; // valuetype AtmRoutingProfileValueType


      interface AtmRoutingProfile_F: itut_x780::ManagedObject_F
      {
/**   Instances of AtmRoutingProfile are created using by the
      AtmRoutingProfile Factory.
**/

            EnumConnectionType connectionTypeGet(in MONameType name)
                  raises (itut_x780::ApplicationError);

            RouteDescriptionListType routeDescriptionListGet(in MONameType name)
                  raises (itut_x780::ApplicationError);

            unsigned short maxHopsGet(in MONameType name)
                  raises (itut_x780::ApplicationError);
```

```
            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectCreation)
            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectDeletion)

      }; // interface AtmRoutingProfile_F


      interface AtmRoutingProfileFactory: itut_x780::ManagedObjectFactory
      {
            itut_x780::ManagedObject create
                  (in NameBindingType nameBinding,
                        // module name containing Name Binding info.
                  in MONameType superior,
                        // Name of containing object.
                  in string reqID,
                        // Requested ID value for name, will be
                        // empty if auto-naming is to be used.
                  out MONameType name,
                        // Entire name of newly created object.
                  in StringSetType packageNameList,
                        // List of packages requested.

                  in EnumConnectionType connectionType,
                  in RouteDescriptionListType routeDescriptionList,
                  in unsigned short maxHops

                  )
                  raises (itut_x780::ApplicationError,
                        itut_x780::CreateError);

      }; // interface AtmRoutingProfileFactory



/**
```

5.4.19 AtmSubnetwork_F

```
*/
      valuetype AtmSubnetworkValueType: itut_m3120::SubnetworkValueType
      {
/**   AtmSubnetwork uses the following inherited from
      itut_m3120:: SubnetworkValueType:
            public SignalIdType       signalId;
                  // GET, SET-BY-CREATE
            public AvailabilityStatusSetType availabilityStatus;
                  // GET
            public SubnetworkNameSetType     containedInSubnetworkList;
                  // GET-REPLACE, (ADD-REMOVE)
            public AbstractLinkEndNameSetTypecontainedLinkEndList;
                  // GET-REPLACE, ADD-REMOVE
            public AbstractLinkNameSetType   containedLinkList;
                  // GET-REPLACE, ADD-REMOVE
            public SubnetworkNameSetType     containedSubnetworkList;
                  // GET-REPLACE, ADD-REMOVE
            public MONameSetType             supportedByObjectList;
                  // GET-REPLACE, ADD-REMOVE
                  // Should provide relationship between subnetwork and
                  // managedElement
            public Istring                   userLabel;
                  // GET-REPLACE
```

```
**/

              public boolean                    opaqueSubnetwork;
                    // True if the Subnetwork is opaque, not allowing the details
                    // of subnetwork connection components to be represented by
                    // object instances.
                    // GET, SET-BY-CREATE

       }; // valuetype AtmSubnetworkValueType



       interface AtmSubnetwork_F: itut_m3120::Subnetwork_F
       {
/**    Instances of AtmSubnetwork are created using by the AtmSubnetworkFactory,
       or automatically by the managed system.
**/

/**    AtmSubnetwork_F inherits the following methods from
       itut_m3120:: Subnetwork_F:
       signalIdGet, availabilityStatusGet, containedInSubnetworkListSet,
       containedInSubnetworkListGet, containedLinkEndListGet, containedLinkEndListSet,
       containedLinkEndListAdd, containedLinkEndListRemove, containedLinkListGet,
       containedLinkListSet, containedLinkListAdd, containedLinkListRemove,
       containedSubnetworkListGet, containedSubnetworkListSet,
       containedSubnetworkListAdd, containedSubnetworkListRemove,
       supportedByObjectListGet, supportedByObjectListSet, supportedByObjectListAdd,
       supportedByObjectListRemove, userLabelGet, userLabelSet

**/

/*     Opaque Subnetwork is TRUE if the Subnetwork is opaque, not allowing the details
       of subnetwork connection components to be represented by object instances.
       The Opaque Subnetwork is still responsible for representing the topology
       elements within the Subnetwork (e.g., contained Subnetworks and Links)
*/
              boolean getOpaqueSubnetwork (in MONameType name)
                    raises (itut_x780::ApplicationError);


/**
The setupPtToPtSNCWithCTP method sets up a point-to-point
connection between non-connected CTPs in the atmSubnetwork.
*/
              void setupPtToPtSNCWithCTP
                    (in MONameType name,
                    in Istring userLabel,
                    in boolean protected,
                    inout AdministrativeStateType adminState,
                    in AtmNetworkCTPNameType aNetworkCTP,
                    in boolean aTrailEndPointInd,
                    in AtmAbstractTrafficDescNameType aTozTrafficDescProfile,
                    in AtmNetworkCTPNameType zNetworkCTP,
                    in boolean zTrailEndPointInd,
                    in AtmAbstractTrafficDescNameType zToaTrafficDescProfile,
                    in AtmRoutingProfileNameType routingProfile,
                    out AtmSNCNameType newSNC)
                    raises (itut_x780::ApplicationError,
                           itut_x780::CreateError,
                           itut_m3120::InvalidTransportServiceCharacteristics,
```

```
                        itut_m3120::IncorrectSubnetworkTerminationPoints,
                        itut_m3120::AEndNetworkTPConnected,
                        itut_m3120::ZEndNetworkTPConnected,
                        itut_m3120::FailureToConnect );

/**
The setupPtToPtSNCWithLinkEnd method sets up a point-to-point
connection between LinkEnds in the atmSubnetwork.
*/
             void setupPtToPtSNCWithLinkEnd
                    (in MONameType name,
                    in Istring userLabel,

                    in boolean protected,
                    inout AdministrativeStateType adminState,
                    in AtmLinkEndNameType aLinkEnd,
                    inout VirtualIDType aVirtualID,
                    in boolean aTrailEndPointInd,
                    in AtmAbstractTrafficDescNameType aTozTrafficDescProfile,
                    in AtmLinkEndNameType zLinkEnd,
                    inout VirtualIDType zVirtualID,
                    in boolean zTrailEndPointInd,
                    in AtmAbstractTrafficDescNameType zToaTrafficDescProfile,
                    in AtmRoutingProfileNameType routingProfile,
                    out AtmSNCNameType newSNC,
                    out AtmNetworkCTPNameType aNetworkCTP,
                    out AtmNetworkCTPNameType zNetworkCTP)
                    raises (itut_x780::ApplicationError,
                           itut_x780::CreateError,
                           itut_m3120::InvalidTransportServiceCharacteristics,
                           itut_m3120::IncorrectSubnetworkTerminationPoints,
                           itut_m3120::AEndNetworkTPConnected,
                           itut_m3120::ZEndNetworkTPConnected,
                           itut_m3120::FailureToConnect);

/**
The setupPtToMultiSNCWithCTP method sets up a composite point-to-
multipoint connection between non-connected CTPs in the atmSubnetwork.
*/
             void setupPtToMultiSNCWithCTP
                    (in MONameType name,
                    in string userLabel,
                    in boolean protected,
                    inout AdministrativeStateType adminState,
                    in AtmNetworkCTPNameType aNetworkCTP,
                    in boolean aTrailEndPointInd,
                    in AtmAbstractTrafficDescNameType aTozTrafficDescProfile,
                    in AtmAbstractTrafficDescNameType aIngressTrafficDescProfile,
                    in ZTPCompositeCtpSetType zTPCompositeCtpSet,
                    in AtmRoutingProfileNameType routingProfile,
                    out AtmSNCNameType newSNC)
                    raises (itut_x780::ApplicationError,
                           itut_x780::CreateError,
                           itut_m3120::InvalidTransportServiceCharacteristics,
                           itut_m3120::IncorrectSubnetworkTerminationPoints,
                           itut_m3120::AEndNetworkTPConnected,
                           itut_m3120::ZEndNetworkTPConnected,
                           itut_m3120::FailureToConnect);

/**
The setupPtToMultiSNCWithLinkEnd method sets up a composite point-to-
multipoint connection between LinkEnds in the atmSubnetwork.
*/
```

```
              void setupPtToMultiSNCWithLinkEnd
                     (in MONameType name,
                     in string userLabel,
                     in boolean protected,
                     inout AdministrativeStateType adminState,
                     in AtmLinkEndNameType aLinkEnd,
                     inout VirtualIDType aVirtualID,
                     in boolean aTrailEndPointInd,
                     in AtmAbstractTrafficDescNameType aTozTrafficDescProfile,
                     in AtmAbstractTrafficDescNameType aIngressTrafficDescProfile,
                     inout ZTPCompositeLinkEndSetType zTPCompositeLinkEndSet,
                     in AtmRoutingProfileNameType routingProfile,
                     out AtmSNCNameType newSNC,
                     out AtmNetworkCTPNameType aNetworkCTP,
                     out AtmNetworkCTPNameType zNetworkCTP )
                     raises (itut_x780::ApplicationError,
                            itut_x780::CreateError,
                            itut_m3120::InvalidTransportServiceCharacteristics,
                            itut_m3120::IncorrectSubnetworkTerminationPoints,
                            itut_m3120::AEndNetworkTPConnected,
                            itut_m3120::ZEndNetworkTPConnected,
                            itut_m3120::FailureToConnect);

/**
The addTpToMultiSNCWithCTP method adds an endpoint to a composite
point-to-multipoint connection in the atmSubnetwork.
*/
              void addTpToMultiSNCWithCTP
                     (in MONameType name,
                     in AtmSNCNameType modifiedSNC,
                     in AtmAbstractTrafficDescNameType aTozTrafficDescProfile,
                     in ZTPCompositeCTPType zTPCompositeCtp,
                     in AtmRoutingProfileNameType routingProfile)
                     raises (itut_x780::ApplicationError,
                            itut_x780::CreateError,
                            itut_m3120::NoSuchSnc,
                            itut_m3120::InvalidTransportServiceCharacteristics,
                            itut_m3120::IncorrectSubnetworkTerminationPoints,
                            itut_m3120::ZEndNetworkTPConnected,
                            itut_m3120::FailureToConnect);

/**
The addTpToMultiSNCWithLinkEnd method adds an endpoint to a
composite point-to-multipoint connection in the atmSubnetwork.
*/
              void addTpToMultiSNCWithLinkEnd
                     (in MONameType name,
                     in AtmSNCNameType modifiedSNC,
                     in AtmAbstractTrafficDescNameType aTozTrafficDescProfile,
                     inout ZTPCompositeLinkEndType zTPCompositeLinkEnd,
                     in AtmRoutingProfileNameType routingProfile,
                     out AtmNetworkCTPNameType zNetworkCTP)
                     raises (itut_x780::ApplicationError,
                            itut_x780::CreateError,
                            itut_m3120::NoSuchSnc,
                            itut_m3120::InvalidTransportServiceCharacteristics,
                            itut_m3120::IncorrectSubnetworkTerminationPoints,
                            itut_m3120::ZEndNetworkTPConnected,
                            itut_m3120::FailureToConnect);

/**
The removeTpFromMultiSNC method removes a TP from a composite
point-to-multipoint connection in the atmSubnetwork.
```

```
*/
              void removeTpFromMultiSNC
                     (in MONameType name,
                     in AtmSNCNameType modifiedSNC,
                     in AtmNetworkCTPNameType ztpRemoved )
                     raises (itut_x780::ApplicationError,
                            itut_m3120::IncorrectSubnetworkTerminationPoints,
                            itut_m3120::NoSuchSnc,
                            itut_m3120::SncConnected,
                            itut_m3120::FailureToRelease);

/**
The releaseSNC method releases a point-to-point or a multipoint connection
between CTPs in the atmSubnetwork.
*/
              void releaseSNC
                     (in MONameType name,
                     in AtmSNCNameType connectionID )
                     raises (itut_x780::ApplicationError,

                            itut_m3120::NoSuchSnc,
                            itut_m3120::SncConnected,
                            itut_m3120::FailureToRelease);

              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, objectCreation)
              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, objectDeletion)
              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, attributeValueChange)

       }; // interface AtmSubnetwork_F


       interface AtmSubnetworkFactory: itut_x780::ManagedObjectFactory
       {
              itut_x780::ManagedObject create
                     (in NameBindingType nameBinding,
                            // module name containing Name Binding info.
                     in MONameType superior,
                            // Name of containing object.
                     in string reqID,
                            // Requested ID value for name, will be
                            // empty if auto-naming is to be used.
                     out MONameType name,
                            // Entire name of newly created object.
                     in StringSetType packageNameList,
                            // List of packages requested.

                     in SignalIdType signalId,
                     in SubnetworkNameSetType containedInSubnetworkList,
                     in AbstractLinkEndNameSetType containedLinkEndList,
                     in AbstractLinkNameSetType containedLinkList,
                     in SubnetworkNameSetType containedSubnetworkList,
                     in MONameSetType supportedByObjectList,
                     in Istring userLabel,
                     in boolean opaqueSubnetwork)
                     raises (itut_x780::ApplicationError,
                            itut_x780::CreateError);

       }; // interface AtmSubnetworkFactory
```

```
/**
```

5.4.20 AtmSNC_F

```
*/
       valuetype AtmSNCValueType: itut_m3120::SNCValueType
       {
/**    AtmSNCValueType uses the following inherited from
       itut_m3120:: PipeValueType:
              public DirectionalityType  directionality;
                     // GET
              public SignalIdType        signalId;
                     // GET, SET-BY-CREATE
              public MONameSetType              aEndNetworkTPList;
                     // GET, SET-BY-CREATE
              public MONameSetType              zEndNetworkTPList;
                     // GET, SET-BY-CREATE
              public AdministrativeStateType    administrativeState;
                     // conditional, present if an instance supports it.
                     // GET-REPLACE
              public OperationalStateType       operationalState;
                     // conditional, present if an instance supports it.
                     // GET
              public AlarmSeverityAssignmentProfileNameType
                     alarmSeverityAssignmentProfilePointer;
                     // conditional, present if the Communications Alarm is
                     // suppprted AND the managed object supports configuration
                     // of alarm severities.
                     // GET-REPLACE
              public AvailabilityStatusSetType availabilityStatus;
                     // conditional, present if an instance supports it.
                     // GET
              public boolean                    protected;
                     // conditional, present if an instance supports it.
                     // GET, SET-BY-CREATE
              public MONameSetType              supportedByObjectList;
                     // conditional, present if an instance supports it.
                     // GET-REPLACE, ADD-REMOVE
              public AlarmStatusType            alarmStatus;
                     // conditional, present if an instance supports
                     // communications alarms.
                     // GET
              public CurrentProblemSetType      currentProblemList;
                     // conditional, present if an instance supports
                     // communications alarms.
                     // GET
              public Istring                    userLabel;
                     // conditional, present if an instance supports it.
                     // GET-REPLACE

       AtmSNCValueType uses the following inherited from
       itut_m3120:: SNCValueType:
              public SNC          compositePointer;
                     // conditional, present if the Sub-network Connection is a
                     // component of another Sub-network Connection within the
                     // same layer (partitioned sub-networks).
                     // GET, SET-BY-CREATE
              public PipeSetType  componentPointerList;
                     // conditional, present if the Sub-network Connection is
                     // made up of a number of component Sub-network
                     // Connections, and Connections, within the same layer
```

```
                  // (partitioned sub-networks).
                  // GET, SET-BY-CREATE
            public MO           relatedRoutingProfile;
                  // conditional, present if routing profiles are supported.
                  // GET, SET-BY-CREATE

**/


            public EnumConnectionType connectionType;
                  // GET, SET-BY-CREATE
            public EnumProvisionType provisionType;
                  // GET-REPLACE
      }; // valuetype AtmSNCValueType


      interface AtmSNC_F: itut_m3120::SNC_F
      {
/**   Instances of AtmSubnetwork are created by the managed system at the
      result of setting up a SNC, using one of the setupSNC methods of
      the AtmSubnetwork object.
**/

/**   AtmSNC_F inherits the following methods from itut_m3120:: Pipe_F:
      directionalityGet, signalIdGet, aEndNetworkTPListGet, zEndNetworkTPListGet,
      administrativeStateGet, administrativeStateSet, operationalStateGet,
      alarmSeverityAssignmentProfilePointerGet,
      alarmSeverityAssignmentProfilePointerSet, availabilityStatusGet, protectedGet,
      supportedByObjectListGet, supportedByObjectListSet, supportedByObjectListAdd,
      supportedByObjectListRemove, alarmStatusGet, currentProblemListGet,
      userLabelGet, userLabelSet

      AtmSNC_F inherits the following methods from itut_m3120:: SNC_F:
      compositePointerGet, componentPointerListGet, relatedRoutingProfileGet


**/



            EnumConnectionType connectionTypeGet
                  (in MONameType name)
                  raises (itut_x780::ApplicationError);

/**
This read/write attribute indicates whether the route for the
associated subnetworkConnection is specified by the administrator
(manual) or determined by the system (automatic) that may include
managing and managed entities of the subnetwork
*/
            EnumProvisionType provisionTypeGet
                  (in MONameType name)
                  raises (itut_x780::ApplicationError);
            void provisionTypeSet
                  (in MONameType name,
                  in EnumProvisionType enumProvision)
                  raises (itut_x780::ApplicationError);


/**
This operation determines the path of the SubnetworkConnection and
returns the path at the lowest possible level supported by the
managed system.  For example, at the lowest level of subnetwork
```

```
partitioning.  The connection trace returns the virtual id (VPI for
VP LND connections or VPI/VCI for VC LND connections) for each
atmLink or external interface point (linkEnd) of the connection.  For
VC connections, the trace should be examined at both the VC level as
well as the VP Level, if both LNDs are under the purview of the
managed system.  In cases of multipoint connection, the results
should be returned in a breadth first fashion.
*/
              ConnTraceSeqType traceSNC
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);

              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, objectCreation)
              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, objectDeletion)
              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, attributeValueChange)
              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, stateChange)
/** SNC instances may generate an ATMSNCReroute alarm whenever the route of
       the SNC changes.  This is useful in cases where the SNC
       represents a soft PVC, or where the EMS provides an opaque network
       that can expose a topology view.
**/
              CONDITIONAL_NOTIFICATION(
                     itut_x780::Notifications, communicationAlarm,
                     tmnCommunicationsAlarmInformationR1Package)


       }; // interface AtmSNC_F

// SNC instances should not be created with a factory
       interface AtmSNCFactory: itut_x780::ManagedObjectFactory
       {

       }; // interface AtmSNCFactory




/**
```

5.4.21 AtmAbstractTrafficDesc_F

```
*/

       valuetype AtmAbstractTrafficDescValueType:
              itut_x780::ManagedObjectValueType
       {
              public EnumServiceCategoryType enumServiceCategory;
                     // GET, SET-BY-CREATE
              public EnumConformanceDefinitionType conformanceDefinition;
                     // GET, SET-BY-CREATE
              public long PeakCellRate;
                     // GET, SET-BY-CREATE
              public long CDVTolerancePCR;
                     // GET, SET-BY-CREATE
                     // conditional

       }; // valuetype AtmAbstractTrafficDescValueType
```

```
      interface AtmAbstractTrafficDesc_F: itut_x780::ManagedObject_F
      {
/**   Instances of ATM Traffic Descriptors are created using by the
      the Factories, or automatically by the managed system.
**/


            EnumServiceCategoryType serviceCategoryGet
                  (in MONameType name)
                  raises (itut_x780::ApplicationError);

            EnumConformanceDefinitionType conformanceDefinitionGet
                  (in MONameType name)
                  raises (itut_x780::ApplicationError);

            long peakCellRateGet
                  (in MONameType name)
                  raises (itut_x780::ApplicationError);

// if policing is performed
            long cDVTolerancePCRGet
                  (in MONameType name)
                  raises (itut_x780::ApplicationError);

            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectCreation)
            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectDeletion)


      }; // interface AtmAbstractTrafficDesc_F



/**
```

5.4.22 AtmTrafficDescABR_F

```
*/
      valuetype AtmTrafficDescABRvalueType: AtmAbstractTrafficDescValueType
      {
            public long aBRMinCellRate;
                  // GET, SET-BY-CREATE
            public long aBRInitialCellRate;
                  // GET, SET-BY-CREATE
            public long aBRTransientBufferExposure;
                  // GET, SET-BY-CREATE
            public EnumABRRateChangeFactorType aBRRateDecreaseFactor;
                  // GET, SET-BY-CREATE
            public EnumABRRateChangeFactorType aBRRateIncreaseFactor;
                  // GET, SET-BY-CREATE
            public long aBRFixedRoundTripTime;
                  // GET, SET-BY-CREATE
            public EnumABRNrmType enumABRNrm;
                  // GET, SET-BY-CREATE
            public EnumABRTrmType enumABRTrm;
                  // GET, SET-BY-CREATE
            public EnumABRCDFType aBRCDF;
                  // GET, SET-BY-CREATE
            public long aBRADTF;
```

```
                // ZERO if not supported
                // GET, SET-BY-CREATE

        }; // valuetype AtmTrafficDescABRvalueType


        interface AtmTrafficDescABR_F: AtmAbstractTrafficDesc_F
        {

                long minCellRateGet
                        (in MONameType name)
                        raises (itut_x780::ApplicationError);

                long initialCellRateGet
                        (in MONameType name)
                        raises (itut_x780::ApplicationError);

                long transientBufferExposureGet
                        (in MONameType name)
                        raises (itut_x780::ApplicationError);

                EnumABRRateChangeFactorType rateDecreaseFactorGet
                        (in MONameType name)
                        raises (itut_x780::ApplicationError);

                EnumABRRateChangeFactorType rateIncreaseFactorGet
                        (in MONameType name)
                        raises (itut_x780::ApplicationError);

                long fixedRoundTripTimeGet
                        (in MONameType name)
                        raises (itut_x780::ApplicationError);

                EnumABRNrmType aBRNrmGet
                        (in MONameType name)
                        raises (itut_x780::ApplicationError);

                EnumABRTrmType aBRTrmGet
                        (in MONameType name)
                        raises (itut_x780::ApplicationError);

                EnumABRCDFType aBRCDFGet
                        (in MONameType name)
                        raises (itut_x780::ApplicationError);

                long aBRADTFGet
                        (in MONameType name)
                        raises (itut_x780::ApplicationError);

                MANDATORY_NOTIFICATION(
                        itut_x780::Notifications, objectCreation)
                MANDATORY_NOTIFICATION(
                        itut_x780::Notifications, objectDeletion)


        }; // interface AtmTrafficDescABR_F


        interface AtmTrafficDescABRFactory: itut_x780::ManagedObjectFactory
        {
                itut_x780::ManagedObject create
                        (in NameBindingType nameBinding,
                                // module name containing Name Binding info.
```

```
                    in MONameType superior,
                           // Name of containing object.
                    in string reqID,
                           // Requested ID value for name, will be
                           // empty if auto-naming is to be used.
                    out MONameType name,
                           // Entire name of newly created object.
                    in StringSetType packageNameList,
                           // List of packages requested.

                    in EnumServiceCategoryType enumServiceCategory,
                    in EnumConformanceDefinitionType conformanceDefinition,
                    in long peakCellRate,
                    in long cDVTolerancePCR,


                    in long aBRMinCellRate,
                    in long aBRInitialCellRate,
                    in long aBRTransientBufferExposure,
                    in EnumABRRateChangeFactorType aBRRateDecreaseFactor,
                    in EnumABRRateChangeFactorType aBRRateIncreaseFactor,
                    in long aBRFixedRoundTripTime,
                    in EnumABRNrmType enumABRNrm,
                    in EnumABRTrmType enumABRTrm,
                    in EnumABRCDFType aBRCDF,
                    in long aBRADTF)

                    raises (itut_x780::ApplicationError,
                           itut_x780::CreateError);

      }; // interface AtmTrafficDescABRFactory



/**
```

5.4.23 AtmTrafficDescCBR_F

```
*/
      valuetype AtmTrafficDescCBRvalueType: AtmAbstractTrafficDescValueType
      {
            public long cBRCLR;
                  // GET, SET-BY-CREATE

      }; // valuetype AtmTrafficDescCBRvalueType


      interface AtmTrafficDescCBR_F: AtmAbstractTrafficDesc_F
      {
            long cBRCLRGet
                  (in MONameType name)
                  raises (itut_x780::ApplicationError);

            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectCreation)
            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectDeletion)


      }; // interface AtmTrafficDescCBR_F
```

```
	interface AtmTrafficDescCBRFactory: itut_x780::ManagedObjectFactory
	{
		itut_x780::ManagedObject create
			(in NameBindingType nameBinding,
				// module name containing Name Binding info.
			in MONameType superior,
				// Name of containing object.
			in string reqID,
				// Requested ID value for name, will be
				// empty if auto-naming is to be used.
			out MONameType name,
				// Entire name of newly created object.
			in StringSetType packageNameList,
				// List of packages requested.

			in EnumServiceCategoryType enumServiceCategory,
			in EnumConformanceDefinitionType conformanceDefinition,
			in long peakCellRate,
			in long cDVTolerancePCR,

			in long cBRCLR)


			raises (itut_x780::ApplicationError,
				itut_x780::CreateError);

	}; // interface AtmTrafficDescCBRFactory



/**
```

5.4.24 AtmTrafficDescVBR_F

```
*/
	valuetype AtmTrafficDescVBRvalueType: AtmAbstractTrafficDescValueType
	{
		public long vBRCDVToleranceSCR;
			// negative if I.371 not supported
			// GET, SET-BY-CREATE
		public long vBRCLR;
			// GET, SET-BY-CREATE
		public long vBRSustainableCellRate;
			// GET, SET-BY-CREATE
		public long vBRMaxBurstSize;
			// GET, SET-BY-CREATE

	}; // valuetype AtmTrafficDescVBRvalueType


	interface AtmTrafficDescVBR_F: AtmAbstractTrafficDesc_F
	{
		long vBRCLRGet
			(in MONameType name)
			raises (itut_x780::ApplicationError);

		long sustainableCellRateGet
			(in MONameType name)
			raises (itut_x780::ApplicationError);

		long maxBurstSizeGet
			(in MONameType name)
```

```
                    raises (itut_x780::ApplicationError);

/**
if policing is performed and If I.371 is supported
*/
              long cDVToleranceSCRGet
                    (in MONameType atmTrafficDescName)
                    raises (itut_x780::ApplicationError);

              MANDATORY_NOTIFICATION(
                    itut_x780::Notifications, objectCreation)
              MANDATORY_NOTIFICATION(
                    itut_x780::Notifications, objectDeletion)


       }; // interface AtmTrafficDescVBR_F


       interface AtmTrafficDescVBRFactory: itut_x780::ManagedObjectFactory
       {
              itut_x780::ManagedObject create
                    (in NameBindingType nameBinding,
                          // module name containing Name Binding info.
                    in MONameType superior,
                          // Name of containing object.
                    in string reqID,
                          // Requested ID value for name, will be
                          // empty if auto-naming is to be used.
                    out MONameType name,
                          // Entire name of newly created object.
                    in StringSetType packageNameList,
                          // List of packages requested.

                    in EnumServiceCategoryType enumServiceCategory,
                    in EnumConformanceDefinitionType conformanceDefinition,
                    in long peakCellRate,
                    in long cDVTolerancePCR,

                    in long vBRCDVToleranceSCR,
                    in long vBRCLR,
                    in long vBRSustainableCellRate,
                    in long vBRMaxBurstSize)


                    raises (itut_x780::ApplicationError,
                          itut_x780::CreateError);

       }; // interface AtmTrafficDescVBRFactory




/**
```

5.4.25 AtmTrafficDescUBR_F

```
*/
       valuetype AtmTrafficDescUBRvalueType: AtmAbstractTrafficDescValueType
       {
       }; // valuetype AtmTrafficDescUBRvalueType


       interface AtmTrafficDescUBR_F: AtmAbstractTrafficDesc_F
```

```
	{
		MANDATORY_NOTIFICATION(
			itut_x780::Notifications, objectCreation)
		MANDATORY_NOTIFICATION(
			itut_x780::Notifications, objectDeletion)


	}; // interface AtmTrafficDescUBR_F


	interface AtmTrafficDescUBRFactory: itut_x780::ManagedObjectFactory
	{
		itut_x780::ManagedObject create
			(in NameBindingType nameBinding,
				// module name containing Name Binding info.
			in MONameType superior,
				// Name of containing object.
			in string reqID,
				// Requested ID value for name, will be
				// empty if auto-naming is to be used.
			out MONameType name,
				// Entire name of newly created object.
			in StringSetType packageNameList,
				// List of packages requested.

			in EnumServiceCategoryType enumServiceCategory,
			in EnumConformanceDefinitionType conformanceDefinition,
			in long peakCellRate,
			in long cDVTolerancePCR)


			raises (itut_x780::ApplicationError,
				itut_x780::CreateError);


	}; // interface AtmTrafficDescUBRFactory



/**
```

5.4.26 AtmTrafficDescGFR_F

```
*/
	valuetype AtmTrafficDescGFRvalueType: AtmAbstractTrafficDescValueType
	{
		public long maxFrameSize;
			// GET, SET-BY-CREATE
		public long minCellRate;
			// for CLP=0
			// GET, SET-BY-CREATE
		public long maxBurstSize;
			// GET, SET-BY-CREATE
		public EnumGFR1or2Type gfrOneOrTwo;
			// GET, SET-BY-CREATE

	}; // valuetype AtmTrafficDescGFRvalueType


	interface AtmTrafficDescGFR_F: AtmAbstractTrafficDesc_F
	{
		long maxFrameSizeGet
```

```
                (in MONameType name)
                raises (itut_x780::ApplicationError);

        long minCellRateGet
                (in MONameType name)
                raises (itut_x780::ApplicationError);

        long maxBurstSizeGet
                (in MONameType name)
                raises (itut_x780::ApplicationError);

        EnumGFR1or2Type gfrOneOrTwoGet
                (in MONameType name)
                raises (itut_x780::ApplicationError);


        MANDATORY_NOTIFICATION(
                itut_x780::Notifications, objectCreation)
        MANDATORY_NOTIFICATION(
                itut_x780::Notifications, objectDeletion)


};  // interface AtmTrafficDescGFR_F


interface AtmTrafficDescGFRFactory: itut_x780::ManagedObjectFactory
{
        itut_x780::ManagedObject create
                (in NameBindingType nameBinding,
                        // module name containing Name Binding info.
                in MONameType superior,
                        // Name of containing object.
                in string reqID,
                        // Requested ID value for name, will be
                        // empty if auto-naming is to be used.
                out MONameType name,
                        // Entire name of newly created object.
                in StringSetType packageNameList,
                        // List of packages requested.

                in EnumServiceCategoryType enumServiceCategory,
                in EnumConformanceDefinitionType conformanceDefinition,
                in long peakCellRate,
                in long cDVTolerancePCR,

                in long maxFrameSize,
                in long minCellRate,
                in long maxBurstSize,
                in EnumGFR1or2Type gfrOneOrTwo)


                raises (itut_x780::ApplicationError,
                        itut_x780::CreateError);

};  // interface AtmTrafficDescGFRFactory


/**
```

5.4.27 CesServiceProfile_F

```
*/
```

```
/**
      This managed object is used to organize data that describes the circuit
      emulation service interworking functions.

      The cesBufferedCDVtolerance attribute specifies the duration of user data that
      must be buffered by the AtmInterworkingTTP managed object to offset Cell Delay
      variation. The recommended default value for DS1 CES is 750 micro seconds and
      1000 micro seconds for DS3 CES.

      The channelAssociatedSignalling attribute identifies which AAL1 format should
      be used. It applies only to structured format. For unstructured format this
      attribute must be set to the default value of basic.

      Instances of this object class shall be explicitly created and deleted by the
      managing system. An instance of this object class shall not be deleted if it is
      in use by any AtmInterworkingTTP object instance.
*/

      valuetype CesServiceProfileValueType: itut_x780::ManagedObjectValueType
      {
            public long  cesBufferedCDVTolerance;
                  // GET, REPLACE
            public EnumChannelAssociatedSignalling channelAssociatedSignalling;
                  // GET, REPLACE

      }; // valuetype CesServiceProfileValueType


      interface CesServiceProfile_F: itut_x780::ManagedObject_F
      {
/*    The Circuit Emulation Service Profile may be created by either a factory
      or by the managed system.  This service profile is pointed to by an
      ATM Network TTP when the termination of the ATM Cell flow provides
      Adaptation to CES
*/

/**   This attribute identifies the duration of user data that must be buffered by
      the AtmInterworkingTTP managed object to offset Cell Delay variation. The
      timing is in increment of 10 micro seconds. The recommended default value for
      DS1 CES is 750 micro seconds and 1000 micro seconds for DS3 CES. The use of
      this attribute is for further study.
*/
            long cesBufferedCDVToleranceGet(in MONameType name)
                  raises (itut_x780::ApplicationError);


/**   This attribute identifies which AAL1 format should be used. This attribute
      applies only to structured format. The default value Basic does not carry
      channel associated signalling (CAS) bits and uses a single 125 usec frame.
      e1Cas, ds1sfCas, and ds1EsfCas carry CAS bits in a multiframe structure for E1,
      DS1 SF, and DS1 ESF respectively.
*/

            EnumChannelAssociatedSignalling channelAssociatedSignallingGet
                  (in MONameType name)
                  raises (itut_x780::ApplicationError);

            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectCreation)
            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectDeletion)

      }; // interface CesServiceProfile_F
```

```
interface CesServiceProfileFactory: itut_x780::ManagedObjectFactory
      {
            itut_x780::ManagedObject create
                  (in NameBindingType nameBinding,
                        // module name containing Name Binding info.
                  in MONameType superior,
                        // Name of containing object.
                  in string reqID,
                        // Requested ID value for name, will be
                        // empty if auto-naming is to be used.
                  out MONameType name,
                        // Entire name of newly created object.
                  in StringSetType packageNameList,
                        // List of packages requested.

                  in long cesBufferedCDVTolerance,
                  in EnumChannelAssociatedSignalling
                        channelAssociatedSignalling)
                  raises (itut_x780::ApplicationError,
                        itut_x780::CreateError);

      }; // interface CesServiceProfileFactory



/**
```

5.4.28 LatestOccurenceLog_F

```
*/

      valuetype LatestOccurrenceLogValueType:
            itut_x780::ManagedObjectValueType
      {
            public AdministrativeStateType adminState;
                  // GET-REPLACE
            public AtmLinkEndNameType linkEndNameType;
                  // GET

      }; // valuetype LatestOccurrenceLogValueType


      interface LatestOccurrenceLog_F : itut_x780::ManagedObject_F
      {
/**   Instances of the LatestOccurrenceLog are created by the managed system
      Specifically, log records shall be automatically created and deleted such
      that for each ATM LinkEnd and Virtual ID pair, only the latest log record
      exists for each type of abnormality.  It is expected that a single instance of
      LatestOccurrenceLog is contained in the top-level subnwork for the EMS.
**/
            AdministrativeStateType administrativeStateGet
                  (in MONameType name)
                  raises (itut_x780::ApplicationError);

            void administrativeStateSet
                  (in MONameType name,
                  in AdministrativeStateType adminState )
                  raises (itut_x780::ApplicationError);

            LatestOccurrenceLogEntrySetType linkEndEntriesGet
```

```
                    (in MONameType name,
                    in AtmLinkEndNameType linkEndNameType)
                    raises (itut_x780::ApplicationError);

      }; // interface LatestOccurrenceLog


      interface LatestOccurrenceLogFactory: itut_x780::ManagedObjectFactory
      {
            itut_x780::ManagedObject create
                    (in NameBindingType nameBinding,
                          // module name containing Name Binding info.
                    in MONameType superior,
                          // Name of containing object.
                    in string reqID,
                          // Requested ID value for name, will be
                          // empty if auto-naming is to be used.
                    out MONameType name,
                          // Entire name of newly created object.
                    in StringSetType packageNameList,
                          // List of packages requested.

                    in AdministrativeStateType adminState)
                    raises (itut_x780::ApplicationError,
                          itut_x780::CreateError);

      }; // interface LatestOccurrenceLogFactory


/**
```

5.4.29 GenericTransportTTP_F

```
*/

      valuetype GenericTransportTTPValueType: itut_m3120:: NetworkTPValueType
      {
/**   GenericTransportTTPValueType uses the following inherited from
      itut_m3120:: TPValueType:
            public MONameSetType                    supportedByObjectList;
                    // points to the supporting circuit pack
                    // GET
            public OperationalStateType             operationalState;
                    // conditional, present if an instance supports it.
                    // GET
            public AlarmStatusType                  alarmStatus;
                    // conditional, present if the TP supports communications
                    // alarm notification.
                    // GET
            public CurrentProblemSetType            currentProblemList;
                    // conditional, present if the TP supports communications
                    // alarm notification.
                    // GET
            public AlarmSeverityAssignmentProfileNameType
                    alarmSeverityAssignmentProfilePointer;
                    // conditional, present if an instance supports
                    // configuration of alarm severities.
                    // GET-REPLACE


      GenericTransportTTPValueType uses the following inherited from
      itut_m3120:: NetworkTPValueType:
            public PointDirectionalityType   pointDirectionality;
```

```
                // GET
            public SignalIdType        signalId;
                // GET, SET-BY-CREATE


**/

            public PortIDType phyTTPPortID;
                // conditional
                // present if the server TTP port is represented
                // GET


            public MONameSetType clientLinkEndPointerList;
                // GET-REPLACE

            public PointCapacityType   potentialCapacity;
                // conditional
                // present if the TTP is a rate adaptive technology
                // GET



      }; // valuetype GenericTransportTTPValueType


      interface GenericTransportTTP_F: itut_m3120::NetworkTP_F
      {
/**   Instances of GenericTransportTTP are created using the
GenericTransportTTPFactory
      or automatically by the managed system.
**/

/**   GenericTransportTTP_F inherits the following methods from
      itut_m3120:: TP_F:
      supportedByObjectListGet,
      operationalStateGet, alarmStatusGet, containedInSubnetworkListGet,
      currentProblemListGet, alarmSeverityAssignmentProfilePointerGet,
      alarmSeverityAssignmentProfilePointerSet

      GenericTransportTTP_F inherits the following methods from
      itut_m3120:: NetworkTP_F:
      pointDirectionalityGet, signalIdGet


**/

            PortIDType portIDGet
                (in MONameType name)
                raises (itut_x780::ApplicationError);


            MONameSetType clientLinkEndPointerListGet
                (in MONameType name)
                raises (itut_x780::ApplicationError);
            void clientLinkEndPointerListSet
                (in MONameType name,
                in MONameSetType clientLinkEndPointerList)
                raises (itut_x780::ApplicationError);

/**

Provides potential bandwidth for rate adaptive server technology.
```

```
*/
            PointCapacityType potentialLinkEndCapacityPackageGet
                  (in MONameType name)
                  raises (itut_x780::ApplicationError);



            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectCreation)
            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectDeletion)
            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, attributeValueChange)
            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, stateChange)
            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, communicationAlarm)



      }; // interface GenericTransportTTP_F



      interface GenericTransportTTPFactory:
            itut_x780::ManagedObjectFactory
      {
            itut_x780::ManagedObject create
                  (in NameBindingType nameBinding,
                        // module name containing Name Binding info.
                  in MONameType superior,
                        // Name of containing object.
                  in string reqID,
                        // Requested ID value for name, will be
                        // empty if auto-naming is to be used.
                  out MONameType name,
                        // Entire name of newly created object.
                  in StringSetType packageNameList,
                        // List of packages requested.

                  in MONameSetType supportedByObjectList,
                        // may be nil
                  in AlarmSeverityAssignmentProfileNameType
                  alarmSeverityAssignmentProfilePointer,
                  in PointDirectionalityType pointDirectionality,

                  in MONameSetType clientLinkEndPointerList)

                  raises (itut_x780::ApplicationError,
                        itut_x780::CreateError);

      }; // interface GenericTransportTTPFactory
```

```
/**
```

5.5. *Interfaces – Fine-grained*

```
The behaviour of the fine-grained interfaces are identical to the corresponding
facade interfaces. The corresponding value types, factories, and object related
comments are defined in the façade interface section.

This section can be omitted from IDL if a management system only supports
facade interfaces.
*/

/**
All fine-grained object inherit the following from X.780:ManagedObject:
       objectClassGet
       packagesGet
       creationSourceGet
       deletePolicyGet
       attributesGet
       destroy
**/


/**
```

5.5.1 AbstractAalProfile

```
*/

       interface AbstractAalProfile: itut_x780::ManagedObject
       {

              EnumAalType  aalTypeGet()
                     raises (itut_x780::ApplicationError);

              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, objectCreation)
              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, objectDeletion)

       }; // interface AbstractAalProfile


/**
```

5.5.2 AalProfileTypeOne

```
*/

       interface AalProfileTypeOne: AbstractAalProfile

       {
              long   cbrRateGet()
                     raises (itut_x780::ApplicationError);

              long   cellLossIntegrationPeriodGet()
                     raises (itut_x780::ApplicationError);

              EnumclockRecoveryType clockRecoveryTypeGet()
                     raises (itut_x780::ApplicationError);


              EnumForwardErrorCorrectionMethod  forwardErrorCorrectionMethodGet()
```

```
                raises (itut_x780::ApplicationError);

        long   partiallyFilledCellsGet()
                raises (itut_x780::ApplicationError);

        boolean      structuredDataTransferGet()
                raises (itut_x780::ApplicationError);

        EnumAal1SubType     subTypeGet()
                raises (itut_x780::ApplicationError);


        MANDATORY_NOTIFICATION(
                itut_x780::Notifications, objectCreation)
        MANDATORY_NOTIFICATION(
                itut_x780::Notifications, objectDeletion)

};  // interface AalProfileTypeOne



/**
```

5.5.3 AalProfileTypeTwo

```
*/

        interface AalProfileTypeTwo: AbstractAalProfile
        {

                long applicationIdentifierGet()
                        raises (itut_x780::ApplicationError);

                EnumAal2ConfigResponsibility     configResponsibilityGet()
                        raises (itut_x780::ApplicationError);

                long   cpsMaxMultiplexedChannelsGet()
                        raises (itut_x780::ApplicationError);

                long   cpsMaxSduLengthGet()
                        raises (itut_x780::ApplicationError);

                long   sscsMaxSssarSduLengthGet()
                        raises (itut_x780::ApplicationError);

                EnumSscsSstedStatus sscsSstedStatusGet()
                        raises (itut_x780::ApplicationError);

                EnumSscsSsadtStatus sscsSsadtStatusGet()
                        raises (itut_x780::ApplicationError);

                EnumSscsServiceCategory sscsServiceCategoryGet()
                        raises (itut_x780::ApplicationError);

                EnumTransportStatus sscsAudioServiceTransportGet()
                        raises (itut_x780::ApplicationError);


                EnumSscsProfileSource sscsProfileSourceGet()
                        raises (itut_x780::ApplicationError);

                long   sscsIeeeOuiGet()
                        raises (itut_x780::ApplicationError);
```

```
            long   SscsPredefinedProfileIdentifierGet()
                   raises (itut_x780::ApplicationError);

            EnumSscsPcmEncoding sscsPcmEncodingGet()
                   raises (itut_x780::ApplicationError);

            EnumTransportStatus sscsFaxDemodulationTransportGet()
                   raises (itut_x780::ApplicationError);

            EnumTransportStatus sscsCasSignalingTransportGet()
                   raises (itut_x780::ApplicationError);

            EnumTransportStatus sscsDtmfDigitPacketTransportGet()
                   raises (itut_x780::ApplicationError);


            EnumTransportStatus sscsMfR1DigitPacketTransportGet()
                   raises (itut_x780::ApplicationError);


            EnumTransportStatus sscsMfR2DigitPacketTransportGet()
                   raises (itut_x780::ApplicationError);

            EnumTransportStatus sscsCircuitModeDataTransportGet()
                   raises (itut_x780::ApplicationError);

            long   sscsCircuitModeDataNumChannelsGet()
                   raises (itut_x780::ApplicationError);

            EnumTransportStatus sscsFrameModeDataTransportGet()
                   raises (itut_x780::ApplicationError);

            long   sscsFrameModeDataMaxLengthGet()
                   raises (itut_x780::ApplicationError);

            long   SscopSduLengthGet()
                   raises (itut_x780::ApplicationError);

            long   sscopUuLengthGet()
                   raises (itut_x780::ApplicationError);


            MANDATORY_NOTIFICATION(
                   itut_x780::Notifications, objectCreation)
            MANDATORY_NOTIFICATION(
                   itut_x780::Notifications, objectDeletion)

      }; // interface AalProfileTypeTwo


/**
```

5.5.4 AalProfileTypeTwoTrunking

```
*/

      interface AalProfileTypeTwoTrunking: AalProfileTypeTwo
      {

            long   vcciGet()
                   raises (itut_x780::ApplicationError);
```

```
              long   signalingVcciGet()
                     raises (itut_x780::ApplicationError);

              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, objectCreation)
              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, objectDeletion)

       }; // interface AalProfileTypeTwoTrunking


/**
```

5.5.5 AalProfileTypeTwoLES

```
*/

interface AalProfileTypeTwoLES: AalProfileTypeTwo
       {

              long   cpsCIDLowerLimitGet()
                     raises (itut_x780::ApplicationError);

              long   cpsCIDUpperLimitGet ()
                     raises (itut_x780::ApplicationError);


              EnumCpsOptimisation cpsOptimisationGet()
                     raises (itut_x780::ApplicationError);

              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, objectCreation)
              MANDATORY_NOTIFICATION(
                     itut_x780::Notifications, objectDeletion)

       }; // interface AalProfileTypeTwoLES




/**
```

5.5.6 AalProfileTypeThreeFour

```
*/

       interface AalProfileTypeThreeFour: AbstractAalProfile

       {
              long   maxCpcsSduSizeForwardGet()
                     raises (itut_x780::ApplicationError);
              long   maxCpcsSduSizeBackwardGet()
                     raises (itut_x780::ApplicationError);


              long   midRangeLowGet()
                     raises (itut_x780::ApplicationError);
              long   midRangeHighGet()
                     raises (itut_x780::ApplicationError);

              EnumAalMode aalModeGet()
                     raises (itut_x780::ApplicationError);
```

```
            EnumSscsType  sscsTypeGet()
                  raises (itut_x780::ApplicationError);

            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectCreation)
            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectDeletion)

      }; // interface AalProfileTypeThreeFour



/**
```

5.5.7 AalProfileTypeFive

```
*/

      interface AalProfileTypeFive: AbstractAalProfile

      {
            long   maxCpcsSduSizeForwardGet()
                  raises (itut_x780::ApplicationError);
            long   maxCpcsSduSizeBackwardGet()
                  raises (itut_x780::ApplicationError);

            EnumAalMode aalModeGet()
                  raises (itut_x780::ApplicationError);

            EnumSscsType  sscsTypeGet()
                  raises (itut_x780::ApplicationError);

            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectCreation)
            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectDeletion)

      }; // interface AalProfileTypeFive



/**
```

5.5.8 AtmImaGroupTP

```
*/

      interface AtmImaGroupTP: itut_m3120::NetworkTP
      {
            MONameSetType associatedImaLinkTPListGet()
                  raises (itut_x780::ApplicationError);
            void associatedImaLinkTPListSet
                  (in MONameSetType associatedImaLinkTPs)
                  raises (itut_x780::ApplicationError);
            void associatedImaLinkTPListAdd
                  (in MONameType newAssociatedImaLinkTP)
                  raises (itut_x780::ApplicationError);
            void associatedImaLinkTPListRemove
                  (in MONameType newAssociatedImaLinkTP)
                  raises (itut_x780::ApplicationError);

            EnumImaGroupState imaGroupNeStateGet()
```

```
        raises (itut_x780::ApplicationError);

    EnumImaGroupState imaGroupFeStateGet()
        raises (itut_x780::ApplicationError);

    EnumImaGroupSymmetry imaGroupSymmetryGet()
        raises (itut_x780::ApplicationError);

    long imaGroupMinNumTxLinksGet()
        raises (itut_x780::ApplicationError);
    void imaGroupMinNumTxLinksSet
        (in long imaGroupMinNumTxLinks)
        raises (itut_x780::ApplicationError);

    long imaGroupMinNumRxLinksGet()
        raises (itut_x780::ApplicationError);
    void imaGroupMinNumRxLinksSet
        (in long imaGroupMinNumRxLinks)
        raises (itut_x780::ApplicationError);

    long imaGroupNumTxCfgLinksGet()
        raises (itut_x780::ApplicationError);

    long imaGroupNumRxCfgLinksGet()
        raises (itut_x780::ApplicationError);

    long imaGroupNumTxActLinksGet()
        raises (itut_x780::ApplicationError);

    long imaGroupNumRxActLinksGet()
        raises (itut_x780::ApplicationError);


    EnumImaGroupTxClkMode imaGroupNeTxClkModeGet()
        raises (itut_x780::ApplicationError);

    void imaGroupNeTxClkModeSet
        (in EnumImaGroupTxClkMode imaGroupNeTxClkMode)
        raises (itut_x780::ApplicationError);

    EnumImaGroupTxClkMode imaGroupFeTxClkModeGet()
        raises (itut_x780::ApplicationError);

    MONameType imaGroupTxTimingRefLinkGet()
        raises (itut_x780::ApplicationError);

    MONameType imaGroupRxTimingRefLinkGet()
        raises (itut_x780::ApplicationError);

    long imaGroupTxImaIdGet()
        raises (itut_x780::ApplicationError);

    long imaGroupRxImaIdGet()
        raises (itut_x780::ApplicationError);

    EnumImaFrameLength imaGroupTxFrameLengthGet()
        raises (itut_x780::ApplicationError);

    void imaGroupTxFrameLengthSet
        (in EnumImaFrameLength imaGroupTxFrameLength)
        raises (itut_x780::ApplicationError);

    EnumImaFrameLength imaGroupRxFrameLengthGet()
```

```
        raises (itut_x780::ApplicationError);

    long imaGroupDiffDelayMaxMSGet()
        raises (itut_x780::ApplicationError);

    long imaGroupDiffDelayMaxMSSet
        (in long imaGroupDiffDelayMaxMS)
        raises (itut_x780::ApplicationError);

    MONameType imaGroupLeastDelayLinkGet()
        raises (itut_x780::ApplicationError);

    long imaGroupDiffDelayMaxObsGet()
        raises (itut_x780::ApplicationError);

    void imaGroupDiffDelayMaxObsReset()
        raises (itut_x780::ApplicationError);

    long imaGroupAlphaValueGet()
        raises (itut_x780::ApplicationError);

    void imaGroupAlphaValueSet
        (in long imaGroupAlphaValue)
        raises (itut_x780::ApplicationError);

    long imaGroupBetaValueGet()
        raises (itut_x780::ApplicationError);

    void imaGroupBetaValueSet
        (in long imaGroupBetaValue)
        raises (itut_x780::ApplicationError);

    long imaGroupGammaValueGet()
        raises (itut_x780::ApplicationError);

    void imaGroupGammaValueSet
        (in long imaGroupGammaValue)
        raises (itut_x780::ApplicationError);

    MONameType imaGroupTestLinkGet()
        raises (itut_x780::ApplicationError);

    void imaGroupTestLinkSet
        (in MONameType imaGroupTestLinkSet)
        raises (itut_x780::ApplicationError);

    long imaGroupTestPatternGet()
        raises (itut_x780::ApplicationError);

    void imaGroupTestPatternSet
        (in long imaGroupTestPattern)
        raises (itut_x780::ApplicationError);

    EnumImaTestProcStatus imaGroupTestProcStatusGet()
        raises (itut_x780::ApplicationError);

    void imaGroupTestProcStatusSet
        (in EnumImaTestProcStatus imaGroupTestProcStatus)
        raises (itut_x780::ApplicationError);


    long imaGroupTxAvailCellRateGet()
        raises (itut_x780::ApplicationError);
```

```
        long imaGroupRxAvailCellRateGet()
                    raises (itut_x780::ApplicationError);



              MANDATORY_NOTIFICATION(
                    itut_x780::Notifications, objectCreation)
              MANDATORY_NOTIFICATION(
                    itut_x780::Notifications, objectDeletion)
              MANDATORY_NOTIFICATION(
                    itut_x780::Notifications, attributeValueChange)
              MANDATORY_NOTIFICATION(
                    itut_x780::Notifications, stateChange)
              MANDATORY_NOTIFICATION(
                    itut_x780::Notifications, communicationAlarm)


        }; // interface AtmImaGroupTP



/**
```

5.5.9 AtmImaLinkTP

```
*/

        interface AtmImaLinkTP: itut_m3120::NetworkTP
        {
              MONameType imaLinkServerTTPGet()
                    raises (itut_x780::ApplicationError);

              MONameType associatedImaGroupTPGet()
                    raises (itut_x780::ApplicationError);
              void associatedImaGroupTPSet
                    (in MONameType associatedImaGroupTP)
                    raises (itut_x780::ApplicationError);

              long imaLinkTxImaIdGet()
                    raises (itut_x780::ApplicationError);

              long imaLinkRxImaIdGet()
                    raises (itut_x780::ApplicationError);

              EnumImaLinkState imaLinkNeTxStateGet()
                    raises (itut_x780::ApplicationError);

              EnumImaLinkState imaLinkFeTxStateGet()
                    raises (itut_x780::ApplicationError);

              EnumImaLinkState imaLinkNeRxStateGet()
                    raises (itut_x780::ApplicationError);

              EnumImaLinkState imaLinkFeRxStateGet()
                    raises (itut_x780::ApplicationError);

              long imaLinkTestPatternGet()
                    raises (itut_x780::ApplicationError);

              void imaLinkTestPatternSet
```

```
                (in long imaLinkTestPattern)
                raises (itut_x780::ApplicationError);

        EnumImaTestProcStatus imaLinkTestProcStatusGet()
                raises (itut_x780::ApplicationError);

        void imaLinkTestProcStatusSet
                (in EnumImaTestProcStatus imaLinkTestProcStatus)
                raises (itut_x780::ApplicationError);

        long imaLinkRelativeDelayGet()
                raises (itut_x780::ApplicationError);



        MANDATORY_NOTIFICATION(
                itut_x780::Notifications, objectCreation)
        MANDATORY_NOTIFICATION(
                itut_x780::Notifications, objectDeletion)
        MANDATORY_NOTIFICATION(
                itut_x780::Notifications, attributeValueChange)
        MANDATORY_NOTIFICATION(
                itut_x780::Notifications, stateChange)
        MANDATORY_NOTIFICATION(
                itut_x780::Notifications, communicationAlarm)


    }; // interface AtmImaLinkTP




/**
```

5.5.10 AtmImaLinkTPPhy

```
*/


    interface AtmImaLinkTPPhy: itut_m3120::NetworkTP
    {

        PortIDType serverTTPPortIDGet()
                raises (itut_x780::ApplicationError);

        void serverTTPPortIDSet
                (in PortIDType serverTTPPortID )
                raises (itut_x780::ApplicationError);

        OperationalStateType serverTTPOpStateGet()
                raises (itut_x780::ApplicationError);

        PointCapacityType potentialCapacityPackageGet()
                raises (itut_x780::ApplicationError);



        MANDATORY_NOTIFICATION(
                itut_x780::Notifications, objectCreation)
        MANDATORY_NOTIFICATION(
                itut_x780::Notifications, objectDeletion)
        MANDATORY_NOTIFICATION(
                itut_x780::Notifications, attributeValueChange)
```

```
            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, stateChange)
            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, communicationAlarm)


      }; // interface AtmImaLinkTPPhy



/**
```

5.5.11 AtmLink

```
*/



      interface AtmLink: itut_m3120::AbstractLink
      {

            AvailabilityStatusSetType availabilityStatusGet()
                  raises (itut_x780::ApplicationError);

            AdministrativeStateType administrativeStateGet()
                  raises (itut_x780::ApplicationError);

            void administrativeStateSet
                  (in AdministrativeStateType adminstrativeState)
                  raises (itut_x780::ApplicationError);

            EnumRestorationModeType restorationModeGet()
                  raises (itut_x780::ApplicationError);

            void restorationModeSet
                  (in EnumRestorationModeType enumRestorationMode)
                  raises (itut_x780::ApplicationError);




            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectCreation)
            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectDeletion)
            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, stateChange)
            CONDITIONAL_NOTIFICATION(
                  itut_x780::Notifications, attributeValueChange,
                  attributeValueChangeNotificationPackage)


      }; // interface AtmLink



/**
```

5.5.12 AtmLinkEnd

```
*/
```

```
interface AtmLinkEnd: itut_m3120::AbstractLinkEnd
{


        SignalIdType signalIdGet()
                raises (itut_x780::ApplicationError);

        AvailabilityStatusSetType availabilityStatusGet()
                raises (itut_x780::ApplicationError);

        AdministrativeStateType administrativeStateGet()
                raises (itut_x780::ApplicationError);

        void administrativeStateSet
                (in AdministrativeStateType adminstrativeState)
                raises (itut_x780::ApplicationError);

        AtmNetworkAccessProfileNameType networkAccessProfileGet()
                raises (itut_x780::ApplicationError);
        void networkAccessProfileSet
                (in AtmNetworkAccessProfileNameType networkAccessProfile)
                raises (itut_x780::ApplicationError);

        EnumLinkEndType linkEndTypeGet()
                raises (itut_x780::ApplicationError);
        void linkEndTypeSet
                (in EnumLinkEndType enumLinkEnd )
                raises (itut_x780::ApplicationError);

        long ingressMaxAssignableBWGet()
                raises (itut_x780::ApplicationError);

        long egressMaxAssignableBWGet()
                raises (itut_x780::ApplicationError);



        MONameSetType serverTTPListGet()
                raises (itut_x780::ApplicationError);

        AtmNetworkCTPNameSeqType supportedCTPsGet()
                raises (itut_x780::ApplicationError);

        void supportedCTPAdd
                (in AtmNetworkCTPNameType supportedCTP )
                raises (itut_x780::ApplicationError);

        void supportedCTPRemove
                (in AtmNetworkCTPNameType supportedCTP )
                raises (itut_x780::ApplicationError);

        LoopbackLocationCodeSeqType loopbackLocIDGet()
                raises (itut_x780::ApplicationError);
        void loopbackLocIDSet
                (in LoopbackLocationCodeSeqType loopbackloc)
                raises (itut_x780::ApplicationError);

        VirtualIDType ilmiVpiVciGet()
                raises (itut_x780::ApplicationError);
        void ilmiVpiVciSet
```

```
        (in VirtualIDType ilmiVpiVci)
        raises (itut_x780::ApplicationError);

long ilmiEstabConnectivityPollIntervalGet()
        raises (itut_x780::ApplicationError);
void ilmiEstabConnectivityPollIntervalSet
        (in long ilmiEstabInt)
        raises (itut_x780::ApplicationError);

long ilmiCheckConnectivityPollIntervalGet()
        raises (itut_x780::ApplicationError);

void ilmiCheckConnectivityPollIntervalSet
        (in long ilmiCheckInt)
        raises (itut_x780::ApplicationError);

long ilmiConnectivityPollFactorGet()
        raises (itut_x780::ApplicationError);

void ilmiConnectivityPollFactorSet
        (in long ilmiPollFactor)
        raises (itut_x780::ApplicationError);

boolean cellScramblingEnabledGet()
        raises (itut_x780::ApplicationError);

void cellScramblingEnabledSet
        (in boolean cellScramblingEnabled)
        raises (itut_x780::ApplicationError);


MONameSetType vendorProfileListGet()
        raises (itut_x780::ApplicationError);

void vendorProfileListAdd
        (in MONameSetType vendorProfileList )
        raises (itut_x780::ApplicationError);

void vendorProfileListRemove
        (in MONameSetType vendorProfileList )
        raises (itut_x780::ApplicationError);


MONameSetType logicalLinkEndListGet()
        raises (itut_x780::ApplicationError);

void logicalLinkEndListSet
        (in MONameSetType logicalLinkEndList)
        raises (itut_x780::ApplicationError);

void logicalLinkEndListAdd
        (in MONameSetType logicalLinkEndList )
        raises (itut_x780::ApplicationError);

void logicalLinkEndListRemove
        (in MONameSetType logicalLinkEndList )
        raises (itut_x780::ApplicationError);

MONameSetType topologicalLinkEndListGet()
        raises (itut_x780::ApplicationError);

void topologicalLinkEndListSet
        (in MONameSetType topologicalLinkEndList)
```

```
                raises (itut_x780::ApplicationError);

        void topologicalLinkEndListAdd
                (in MONameSetType topologicalLinkEndList )
                raises (itut_x780::ApplicationError);

        void topologicalLinkEndListRemove
                (in MONameSetType topologicalLinkEndList )
                raises (itut_x780::ApplicationError);

        void linkPVCTrace
                (in EnumLinkTraceType enumLinkTrace,
                in AtmSubnetworkNameSeqType selectedSubnets,
                out AtmSNCNameSeqType traceResults)
                raises (itut_x780::ApplicationError);

        MANDATORY_NOTIFICATION(
                itut_x780::Notifications, objectCreation)
        MANDATORY_NOTIFICATION(
                itut_x780::Notifications, objectDeletion)
        MANDATORY_NOTIFICATION(
                itut_x780::Notifications, attributeValueChange)
        MANDATORY_NOTIFICATION(
                itut_x780::Notifications, stateChange)

    }; // interface AtmLinkEnd




/**
```

5.5.13 AtmLinkEndPhy

```
*/


    interface AtmLinkEndPhy:AtmLinkEnd
    {

        CharacteristicInfoType serverTTPCharInfoGet()
                raises (itut_x780::ApplicationError);

        PortIDType serverTTPPortIDGet()
                raises (itut_x780::ApplicationError);

        void serverTTPPortIDSet
                (in PortIDType serverTTPPortID )
                raises (itut_x780::ApplicationError);

        OperationalStateType serverTTPOpStateGet()
                raises (itut_x780::ApplicationError);


        AlarmSeverityAssignmentProfileNameType profileGet()
                raises (itut_x780::ApplicationError);

        AlarmSeverityAssignmentProfileNameType profileSet
                (in AlarmSeverityAssignmentProfileNameType asapName)
                raises (itut_x780::ApplicationError);
```

```
            CurrentProblemSetType currentProblemListGet()
                  raises (itut_x780::ApplicationError);

            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectCreation)
            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectDeletion)
            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, attributeValueChange)
            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, stateChange)
            CONDITIONAL_NOTIFICATION(
                  itut_x780::Notifications, communicationAlarm,
                  tmnCommunicationsAlarmInformationR1Package)



      }; // interface AtmLinkEndPhy




/**
```

5.5.14 AtmNetworkAccessProfile

```
*/


      interface AtmNetworkAccessProfile: itut_x780::ManagedObject
      {
            long totalIngressBWGet()
                  raises (itut_x780::ApplicationError);

            long totalEgressBWGet()
                  raises (itut_x780::ApplicationError);


            long maxNumActiveVcConnGet()
                  raises (itut_x780::ApplicationError);


            long maxNumActiveVpConnGet()
                  raises (itut_x780::ApplicationError);


            VpiOrVciRangeType vpiRangeGet()
                  raises (itut_x780::ApplicationError);


            VpiOrVciRangeType vciRangeGet()
                  raises (itut_x780::ApplicationError);


            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectCreation)
            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectDeletion)
```

```
	}; // interface AtmNetworkAccessProfile




/**
```

5.5.15 AtmNetworkTP

```
*/


	interface AtmNetworkTP: itut_m3120::NetworkTP
	{


		VirtualIDType networkTPVpiVciGet()
			raises (itut_x780::ApplicationError);


		LoopbackCellReplyType loopbackOamCell
			(in LoopbackLocType loopbackLoc,
			in EnumOamCellType oamCellType )
			raises (itut_x780::ApplicationError);

		EnumPmOamMethodType pmOamMethodGet()
			raises (itut_x780::ApplicationError);

		void pmOamMethodSet
			(in EnumPmOamMethodType enumPmOamMethod )
			raises (itut_x780::ApplicationError);

		EnumPmOamDirectionType pmOamDirectionGet()
			raises (itut_x780::ApplicationError);

		void pmOamDirectionSet
			(in EnumPmOamDirectionType enumPmOamDirection )
			raises (itut_x780::ApplicationError);

		EnumPmOamBlockSizeType pmOamBlockSizeGet()
			raises (itut_x780::ApplicationError);

		void pmOamBlockSizeSet
			(in EnumPmOamBlockSizeType pmBlockSize)
			raises (itut_x780::ApplicationError);

		boolean pmOamForwardActiveGet()
			raises (itut_x780::ApplicationError);

		void pmOamForwardActiveSet
			(in boolean pmForwardActive )
			raises (itut_x780::ApplicationError);

		boolean pmOamBackwardActiveGet()
			raises (itut_x780::ApplicationError);

		void pmOamBackwardActiveSet
			(in boolean pmBackwardActive )
			raises (itut_x780::ApplicationError);
```

```
            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectCreation)
            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectDeletion)
            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, attributeValueChange)
            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, stateChange)


      }; // interface AtmNetworkTP


/**
```

5.5.16 AtmNetworkCTP

```
*/


      interface AtmNetworkCTP: AtmNetworkTP
      {

            boolean segmentEndpointGet()
                  raises (itut_x780::ApplicationError);
            void segmentEndpointSet
                  (in boolean segmentEndpoint)
                  raises (itut_x780::ApplicationError);

            AtmNetworkTTPNameType relatedAtmTTPGet()
                  raises (itut_x780::ApplicationError);
            void relatedAtmTTPSet
                  (in AtmNetworkTTPNameType relatedAtmTTP)
                  raises (itut_x780::ApplicationError);


            AtmAbstractTrafficDescNameType egressTrafficDescProfileGet()
                  raises (itut_x780::ApplicationError);

            void egressTrafficDescProfileSet
                  (in AtmAbstractTrafficDescNameType egressProfile)
                  raises (itut_x780::ApplicationError);

            AtmAbstractTrafficDescNameType ingressTrafficDescProfileGet()
                  raises (itut_x780::ApplicationError);

            void ingressTrafficDescProfileSet
                  (in AtmAbstractTrafficDescNameType ingressProfile)
                  raises (itut_x780::ApplicationError);


            boolean ingressTaggingIndGet()
                  raises (itut_x780::ApplicationError);

            void ingressTaggingIndSet
                  (in boolean ingressTagInd)
                  raises (itut_x780::ApplicationError);

            boolean egressTaggingIndGet()
                  raises (itut_x780::ApplicationError);

            void egressTaggingIndSet
                  (in boolean egressTagInd)
```

```
                raises (itut_x780::ApplicationError);


            MANDATORY_NOTIFICATION(
                itut_x780::Notifications, objectCreation)
            MANDATORY_NOTIFICATION(
                itut_x780::Notifications, objectDeletion)
            MANDATORY_NOTIFICATION(
                itut_x780::Notifications, attributeValueChange)
            MANDATORY_NOTIFICATION(
                itut_x780::Notifications, stateChange)
            CONDITIONAL_NOTIFICATION(
                itut_x780::Notifications, communicationAlarm,
                tmnCommunicationsAlarmInformationR1Package)


      }; // interface AtmNetworkCTP




/**
```

5.5.17 AtmNetworkTTP

```
*/


      interface AtmNetworkTTP: AtmNetworkTP
      {

            AtmNetworkCTPNameType relatedAtmCTPGet()
                raises (itut_x780::ApplicationError);

            MONameSetType clientLinkEndPointerListGet()
                raises (itut_x780::ApplicationError);
            void clientLinkEndPointerListSet
                (in MONameSetType clientLinkEndPointerList)
                raises (itut_x780::ApplicationError);

            MONameType serviceProfilePointerGet()
                raises (itut_x780::ApplicationError);
            void serviceProfilePointerSet(in MONameType serviceProfilePointer)
                raises (itut_x780::ApplicationError);


            MONameType aalProfilePointerGet()
                raises (itut_x780::ApplicationError);
            void AalProfilePointerSet(
                in MONameType aalProfilePointer)
                raises (itut_x780::ApplicationError);


            MANDATORY_NOTIFICATION(
                itut_x780::Notifications, objectCreation)
            MANDATORY_NOTIFICATION(
                itut_x780::Notifications, objectDeletion)
            MANDATORY_NOTIFICATION(
                itut_x780::Notifications, attributeValueChange)
            MANDATORY_NOTIFICATION(
```

```
                itut_x780::Notifications, stateChange)
        CONDITIONAL_NOTIFICATION(
                itut_x780::Notifications, communicationAlarm,
                tmnCommunicationsAlarmInformationR1Package)


    }; // interface AtmNetworkTTP




/**
```

5.5.18 AtmRoutingProfile

```
*/

    interface AtmRoutingProfile: itut_x780::ManagedObject
    {

        EnumConnectionType connectionTypeGet()
            raises (itut_x780::ApplicationError);

        RouteDescriptionListType routeDescriptionListGet()
            raises (itut_x780::ApplicationError);

        unsigned short maxHopsGet()
            raises (itut_x780::ApplicationError);

        MANDATORY_NOTIFICATION(
            itut_x780::Notifications, objectCreation)
        MANDATORY_NOTIFICATION(
            itut_x780::Notifications, objectDeletion)

    }; // interface AtmRoutingProfile




/**
```

5.5.19 AtmSubnetwork

```
*/


    interface AtmSubnetwork: itut_m3120::Subnetwork
    {


        boolean getOpaqueSubnetwork ()
            raises (itut_x780::ApplicationError);


        void setupPtToPtSNCWithCTP
            (
            in Istring userLabel,
            in boolean protected,
```

```
                inout AdministrativeStateType adminState,
                in AtmNetworkCTPNameType aNetworkCTP,
                in boolean aTrailEndPointInd,
                in AtmAbstractTrafficDescNameType aTozTrafficDescProfile,
                in AtmNetworkCTPNameType zNetworkCTP,
                in boolean zTrailEndPointInd,
                in AtmAbstractTrafficDescNameType zToaTrafficDescProfile,
                in AtmRoutingProfileNameType routingProfile,
                out AtmSNCNameType newSNC)
                raises (itut_x780::ApplicationError,
                       itut_x780::CreateError,
                       itut_m3120::InvalidTransportServiceCharacteristics,
                       itut_m3120::IncorrectSubnetworkTerminationPoints,
                       itut_m3120::AEndNetworkTPConnected,
                       itut_m3120::ZEndNetworkTPConnected,
                       itut_m3120::FailureToConnect );

        void setupPtToPtSNCWithLinkEnd
                (
                in Istring userLabel,

                in boolean protected,
                inout AdministrativeStateType adminState,
                in AtmLinkEndNameType aLinkEnd,
                inout VirtualIDType aVirtualID,
                in boolean aTrailEndPointInd,
                in AtmAbstractTrafficDescNameType aTozTrafficDescProfile,
                in AtmLinkEndNameType zLinkEnd,
                inout VirtualIDType zVirtualID,
                in boolean zTrailEndPointInd,
                in AtmAbstractTrafficDescNameType zToaTrafficDescProfile,
                in AtmRoutingProfileNameType routingProfile,
                out AtmSNCNameType newSNC,
                out AtmNetworkCTPNameType aNetworkCTP,
                out AtmNetworkCTPNameType zNetworkCTP)
                raises (itut_x780::ApplicationError,
                       itut_x780::CreateError,
                       itut_m3120::InvalidTransportServiceCharacteristics,
                       itut_m3120::IncorrectSubnetworkTerminationPoints,
                       itut_m3120::AEndNetworkTPConnected,
                       itut_m3120::ZEndNetworkTPConnected,
                       itut_m3120::FailureToConnect);

        void setupPtToMultiSNCWithCTP
                (
                in string userLabel,
                in boolean protected,
                inout AdministrativeStateType adminState,
                in AtmNetworkCTPNameType aNetworkCTP,
                in boolean aTrailEndPointInd,
                in AtmAbstractTrafficDescNameType aTozTrafficDescProfile,
                in AtmAbstractTrafficDescNameType aIngressTrafficDescProfile,
                in ZTPCompositeCtpSetType zTPCompositeCtpSet,
                in AtmRoutingProfileNameType routingProfile,
                out AtmSNCNameType newSNC)
                raises (itut_x780::ApplicationError,
                       itut_x780::CreateError,
                       itut_m3120::InvalidTransportServiceCharacteristics,
                       itut_m3120::IncorrectSubnetworkTerminationPoints,
                       itut_m3120::AEndNetworkTPConnected,
                       itut_m3120::ZEndNetworkTPConnected,
                       itut_m3120::FailureToConnect);
```

```
void setupPtToMultiSNCWithLinkEnd
      (
      in string userLabel,
      in boolean protected,
      inout AdministrativeStateType adminState,
      in AtmLinkEndNameType aLinkEnd,
      inout VirtualIDType aVirtualID,
      in boolean aTrailEndPointInd,
      in AtmAbstractTrafficDescNameType aTozTrafficDescProfile,
      in AtmAbstractTrafficDescNameType aIngressTrafficDescProfile,
      inout ZTPCompositeLinkEndSetType zTPCompositeLinkEndSet,
      in AtmRoutingProfileNameType routingProfile,
      out AtmSNCNameType newSNC,
      out AtmNetworkCTPNameType aNetworkCTP,
      out AtmNetworkCTPNameType zNetworkCTP )
      raises (itut_x780::ApplicationError,
            itut_x780::CreateError,
            itut_m3120::InvalidTransportServiceCharacteristics,
            itut_m3120::IncorrectSubnetworkTerminationPoints,
            itut_m3120::AEndNetworkTPConnected,
            itut_m3120::ZEndNetworkTPConnected,
            itut_m3120::FailureToConnect);

void addTpToMultiSNCWithCTP
      (
      in AtmSNCNameType modifiedSNC,
      in AtmAbstractTrafficDescNameType aTozTrafficDescProfile,
      in ZTPCompositeCTPType zTPCompositeCtp,
      in AtmRoutingProfileNameType routingProfile)
      raises (itut_x780::ApplicationError,
            itut_x780::CreateError,
            itut_m3120::NoSuchSnc,
            itut_m3120::InvalidTransportServiceCharacteristics,
            itut_m3120::IncorrectSubnetworkTerminationPoints,
            itut_m3120::ZEndNetworkTPConnected,
            itut_m3120::FailureToConnect);

void addTpToMultiSNCWithLinkEnd
      (
      in AtmSNCNameType modifiedSNC,
      in AtmAbstractTrafficDescNameType aTozTrafficDescProfile,
      inout ZTPCompositeLinkEndType zTPCompositeLinkEnd,
      in AtmRoutingProfileNameType routingProfile,
      out AtmNetworkCTPNameType zNetworkCTP)
      raises (itut_x780::ApplicationError,
            itut_x780::CreateError,
            itut_m3120::NoSuchSnc,
            itut_m3120::InvalidTransportServiceCharacteristics,
            itut_m3120::IncorrectSubnetworkTerminationPoints,
            itut_m3120::ZEndNetworkTPConnected,
            itut_m3120::FailureToConnect);

void removeTpFromMultiSNC
      (
      in AtmSNCNameType modifiedSNC,
      in AtmNetworkCTPNameType ztpRemoved )
      raises (itut_x780::ApplicationError,
            itut_m3120::IncorrectSubnetworkTerminationPoints,
            itut_m3120::NoSuchSnc,
            itut_m3120::SncConnected,
            itut_m3120::FailureToRelease);

void releaseSNC
```

```
                    (
                    in AtmSNCNameType connectionID )
                    raises (itut_x780::ApplicationError,

                          itut_m3120::NoSuchSnc,
                          itut_m3120::SncConnected,
                          itut_m3120::FailureToRelease);

             MANDATORY_NOTIFICATION(
                    itut_x780::Notifications, objectCreation)
             MANDATORY_NOTIFICATION(
                    itut_x780::Notifications, objectDeletion)
             MANDATORY_NOTIFICATION(
                    itut_x780::Notifications, attributeValueChange)

      }; // interface AtmSubnetwork




/**
```

5.5.20 AtmSNC

```
*/


      interface AtmSNC: itut_m3120::SNC
      {


             EnumConnectionType connectionTypeGet()
                    raises (itut_x780::ApplicationError);

             EnumProvisionType provisionTypeGet()
                    raises (itut_x780::ApplicationError);
             void provisionTypeSet
                    (in EnumProvisionType enumProvision)
                    raises (itut_x780::ApplicationError);


             ConnTraceSeqType traceSNC()
                    raises (itut_x780::ApplicationError);

             MANDATORY_NOTIFICATION(
                    itut_x780::Notifications, objectCreation)
             MANDATORY_NOTIFICATION(
                    itut_x780::Notifications, objectDeletion)
             MANDATORY_NOTIFICATION(
                    itut_x780::Notifications, attributeValueChange)
             MANDATORY_NOTIFICATION(
                    itut_x780::Notifications, stateChange)
             CONDITIONAL_NOTIFICATION(
                    itut_x780::Notifications, communicationAlarm,
                    tmnCommunicationsAlarmInformationR1Package)


      }; // interface AtmSNC
```

```
/**
```

5.5.21 AtmAbstractTrafficDesc

```
*/


      interface AtmAbstractTrafficDesc: itut_x780::ManagedObject
      {


            EnumServiceCategoryType serviceCategoryGet()
                  raises (itut_x780::ApplicationError);

            EnumConformanceDefinitionType conformanceDefinitionGet()
                  raises (itut_x780::ApplicationError);

            long peakCellRateGet()
                  raises (itut_x780::ApplicationError);

            long cDVTolerancePCRGet()
                  raises (itut_x780::ApplicationError);

            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectCreation)
            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectDeletion)


      }; // interface AtmAbstractTrafficDesc



/**
```

5.5.22 AtmTrafficDescABR

```
*/


      interface AtmTrafficDescABR: AtmAbstractTrafficDesc
      {

            long minCellRateGet()
                  raises (itut_x780::ApplicationError);

            long initialCellRateGet()
                  raises (itut_x780::ApplicationError);

            long transientBufferExposureGet()
                  raises (itut_x780::ApplicationError);

            EnumABRRateChangeFactorType rateDecreaseFactorGet()
                  raises (itut_x780::ApplicationError);

            EnumABRRateChangeFactorType rateIncreaseFactorGet()
                  raises (itut_x780::ApplicationError);
```

```
            long fixedRoundTripTimeGet()
                  raises (itut_x780::ApplicationError);

            EnumABRNrmType aBRNrmGet()
                  raises (itut_x780::ApplicationError);

            EnumABRTrmType aBRTrmGet()
                  raises (itut_x780::ApplicationError);

            EnumABRCDFType aBRCDFGet()
                  raises (itut_x780::ApplicationError);

            long aBRADTFGet()
                  raises (itut_x780::ApplicationError);

            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectCreation)
            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectDeletion)


      }; // interface AtmTrafficDescABR




/**
```

5.5.23 AtmTrafficDescCBR

```
*/


      interface AtmTrafficDescCBR: AtmAbstractTrafficDesc
      {
            long cBRCLRGet()
                  raises (itut_x780::ApplicationError);

            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectCreation)
            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectDeletion)


      }; // interface AtmTrafficDescCBR




/**
```

5.5.24 AtmTrafficDescVBR

```
*/


      interface AtmTrafficDescVBR: AtmAbstractTrafficDesc
      {
```

```
            long vBRCLRGet()
                  raises (itut_x780::ApplicationError);

            long sustainableCellRateGet()
                  raises (itut_x780::ApplicationError);

            long maxBurstSizeGet()
                  raises (itut_x780::ApplicationError);

            long cDVToleranceSCRGet()
                  raises (itut_x780::ApplicationError);

            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectCreation)
            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectDeletion)


      }; // interface AtmTrafficDescVBR




/**
```

5.5.25 AtmTrafficDescUBR

```
*/


      interface AtmTrafficDescUBR: AtmAbstractTrafficDesc
      {
            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectCreation)
            MANDATORY_NOTIFICATION(
                  itut_x780::Notifications, objectDeletion)


      }; // interface AtmTrafficDescUBR




/**
```

5.5.26 AtmTrafficDescGFR

```
*/


      interface AtmTrafficDescGFR: AtmAbstractTrafficDesc
      {
            long maxFrameSizeGet()
                  raises (itut_x780::ApplicationError);

            long minCellRateGet()
                  raises (itut_x780::ApplicationError);

            long maxBurstSizeGet()
                  raises (itut_x780::ApplicationError);

            EnumGFR1or2Type gfrOneOrTwoGet()
```

```
                raises (itut_x780::ApplicationError);


        MANDATORY_NOTIFICATION(
                itut_x780::Notifications, objectCreation)
        MANDATORY_NOTIFICATION(
                itut_x780::Notifications, objectDeletion)


    }; // interface AtmTrafficDescGFR




/**
```

5.5.27 CesServiceProfile

```
*/


    interface CesServiceProfile: itut_x780::ManagedObject
    {
        long cesBufferedCDVToleranceGet()
                raises (itut_x780::ApplicationError);



        EnumChannelAssociatedSignalling channelAssociatedSignallingGet()
                raises (itut_x780::ApplicationError);

        MANDATORY_NOTIFICATION(
                itut_x780::Notifications, objectCreation)
        MANDATORY_NOTIFICATION(
                itut_x780::Notifications, objectDeletion)

    }; // interface CesServiceProfile




/**
```

5.5.28 LatestOccurenceLog

```
*/


    interface LatestOccurrenceLog : itut_x780::ManagedObject
    {
        AdministrativeStateType administrativeStateGet()
                raises (itut_x780::ApplicationError);

        void administrativeStateSet
                (in AdministrativeStateType adminState )
                raises (itut_x780::ApplicationError);

        LatestOccurrenceLogEntrySetType linkEndEntriesGet
```

```
                (in AtmLinkEndNameType linkEndNameType)
                raises (itut_x780::ApplicationError);

        }; // interface LatestOccurrenceLog


/**
```

5.5.29 GenericTransportTTP

```
*/


        interface GenericTransportTTP: itut_m3120::NetworkTP
        {


                PortIDType portIDGet()
                        raises (itut_x780::ApplicationError);


                MONameSetType clientLinkEndPointerListGet()
                        raises (itut_x780::ApplicationError);
                void clientLinkEndPointerListSet
                        (in MONameSetType clientLinkEndPointerList)
                        raises (itut_x780::ApplicationError);

                PointCapacityType potentialLinkEndCapacityPackageGet()
                        raises (itut_x780::ApplicationError);



                MANDATORY_NOTIFICATION(
                        itut_x780::Notifications, objectCreation)
                MANDATORY_NOTIFICATION(
                        itut_x780::Notifications, objectDeletion)
                MANDATORY_NOTIFICATION(
                        itut_x780::Notifications, attributeValueChange)
                MANDATORY_NOTIFICATION(
                        itut_x780::Notifications, stateChange)
                MANDATORY_NOTIFICATION(
                        itut_x780::Notifications, communicationAlarm)



        }; // interface GenericTransportTTP
```

```
/**
```

5.6. *Performance Management – Facade*

```
This section defines the interfaces for supporting Performance Management (PM)
for ATM. Further discussion of PM-related managed entities and attributes is
provided in Sections 3.17, 3.18, 3.26, 3.26. 3.41, 3.42, 3.90, 3.91 and 3.92
of the ATM Network View Requirements and Logical MIB.  All of the performance
management aspects for module atmf_m4nw_v2 are grouped into this section.
It draws upon the technology independent performance management module in
ITU-T Rec. Q.822.1 and adds extensions that are specific to ATM.

The behaviour of the facade interfaces are identical to the corresponding
fine-grained interfaces. Therefore, comments are not included in the façade
interfaces. Readers are referred to the fine-grained interface in Section 4.6
for the behaviour of the facade interface.

This section can be omitted from IDL if a management system only supports
fine-grained interface.
*/

/**
Interfaces imported from itut_q822d1:

CurrentData
*/

/*
```

5.6.1 Forward Declarations

```
*/



/**
Typedef forward declarations
*/
       typedef MONameType CellProtocolMonCDNameType;
       typedef MONameType CongDiscardCDNameType;
       typedef MONameType PmOamCDNameType;
       typedef MONameType TcAdaptProtMonCDNameType;
       typedef MONameType TrafficLoadCDNameType;
       typedef MONameType UpcNpcDisagreementsCDNameType;




/**
```

5.6.2 CellProtocolMonCD_F (Cell Protocol Monitoring Current Data)

```
Retrieves attributes or current and history data counter values for counters
within the Cell Protocol Monitoring grouping, namely, numberDiscCellsProtErr
and numberRecvOAMCells.

Additional inherited methods from Scanner provide for the setting of
AdministrativeState, retrieval of OperationalState, and setting of Granularity
Period.  Inherited methods from CurrentData provide for retrieval of the
SuspectFlag, retrieval of the ElapsedTime, activating or deactivating
HistoryRetention, associating threshold data with a current data instance, and
```

```
activating or deactivating the suppression of counters having all-zero counts.

Applicable methods from Scanner include: administrativeStateGet,
administrativeStateSet, operationalStateGet, granularityPeriodGet,
and granularityPeriodSet.

Applicable methods from CurrentData include: suspectIntervalFlagGet,
suspectIntervalFlagDefaultSet, elapsedTimeGet, historyRetentionGet,
historyRetentionSet, thresholdDataInstanceListGet, thresholdDataInstanceListSet,
getMostRecent, getBetween, numSuppressedIntervalsGet,
maxSuppressedIntervalsGet, maxSuppressedIntervalsSet.

This object should support the following notifications: objectCreation,
objectDeletion, attributeValueChange, and stateChange.

A Threshold Crossing Alert (qualityOfServiceAlarm) is used to notify the
management system when any of the performance parameters exceeds a pre-set
threshold described in the associated ThresholdData object.

A containment relationship exists between an AtmLinkEnd and its associated
CellProtocolMonCurrentData object.
*/
       valuetype CellProtocolMonCDValueType:
              itut_q822d1::CurrentDataValueType
       {
              public unsigned long numberDiscCellsProtErr;
                     // GET
              public unsigned long numberRecvOAMCells;
                     // GET
       }; // valuetype CellProtocolMonCDValueType


       valuetype CellProtocolMonHDValueType: itut_q822d1::HistoryDataValueType
       {
              public unsigned long numberDiscCellsProtErr;
                     // GET
              public unsigned long numberRecvOAMCells;
                     // GET
       }; // valuetype CellProtocolMonHistoryDataValueType


       interface CellProtocolMonCD_F: itut_q822d1::CurrentData_F
       {
/**
Count of cells discarded due to protocol errors; thresholded count.
*/
              unsigned long numberDiscCellsProtErrGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);
/**
Count of the number of received OAM Cells.
*/
              unsigned long numberRecvOAMCellsGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);

       }; // interface CellProtocolMonCD_F


       interface CellProtocolMonCDFactory: itut_x780::ManagedObjectFactory
       {
              itut_x780::ManagedObject create
                     (in NameBindingType nameBinding,
```

```
                        // module name containing Name Binding info.
                    in MONameType superior,
                        // Name of containing object.
                    in string reqID,
                        // Requested ID value for name, will be
                        // empty if auto-naming is to be used.
                    out MONameType name,
                        // Entire name of newly created object.
                    in StringSetType packageNameList,
                        // List of packages requested.

                    in short historyRetention,
                    in AdministrativeStateType administrativeState,
                    in TimePeriodType granularityPeriod,
                    in MONameSetType thresholdDataInstanceList,
                    in short maxSuppressedIntervals,
                    in GeneralizedTimeType periodSynchronizationTime)
                    raises (itut_x780::ApplicationError,
                        itut_x780::CreateError);

    }; // interface CellProtocolMonCDFactory



/**
```

5.6.3 CongDiscardCD_F (ATM Congestion Discards Current Data)

```
Retrieves attributes or current and history data counter values for counters within
the ATM Congestion Discards grouping, namely, numberCellsDiscCong and
numberCLP0CellsDisc.

Additional inherited methods from Scanner provide for the setting of
AdministrativeState, retrieval of OperationalState, and setting of Granularity Period.
Inherited methods from CurrentData provide for retrieval of the SuspectFlag, retrieval
of the ElapsedTime, activating or deactivating HistoryRetention, associating threshold
data with a current data instance, and activating or deactivating the suppression of
counters having all-zero counts.

Applicable methods from Scanner include: administrativeStateGet,
administrativeStateSet, operationalStateGet, granularityPeriodGet, and
granularityPeriodSet.

Applicable methods from CurrentData include: suspectIntervalFlagGet,
suspectIntervalFlagDefaultSet, elapsedTimeGet, historyRetentionGet,
historyRetentionSet, thresholdDataInstanceListGet, thresholdDataInstanceListSet,
getMostRecent, getBetween, numSuppressedIntervalsGet, maxSuppressedIntervalsGet,
maxSuppressedIntervalsSet.

This object should support the following notifications: objectCreation,
objectDeletion, attributeValueChange, and stateChange.

A Threshold Crossing Alert (qualityOfServiceAlarm) is used to notify the management
system when any of the performance parameters exceeds a pre-set threshold described in
the associated ThresholdData object.

A containment relationship exists between an AtmLinkEnd and its associated
CongDiscardCurrentData object.
*/
    valuetype CongDiscardCDValueType: itut_q822d1::CurrentDataValueType
    {
```

```
              public unsigned long numberCellsDiscCong;
                     // GET
              public unsigned long numberCLP0CellsDisc;
                     // GET
       }; // valuetype CongDiscardCDValueType


       valuetype CongDiscardHDValueType: itut_q822d1::HistoryDataValueType
       {
              public unsigned long numberCellsDiscCong;
                     // GET
              public unsigned long numberCLP0CellsDisc;
                     // GET
       }; // valuetype CongDiscardHDValueType


       interface CongDiscardCD_F: itut_q822d1::CurrentData_F
       {
/**
Count of cells discarded due to congestion; thresholded count.
*/
              unsigned long numberCellsDiscCongGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);

/**
Count of high prioritycells discarded due to congestion; thresholded count.
*/
              unsigned long numberCLP0CellsDiscGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);
       }; // interface CongDiscardCD_F


       interface CongDiscardCDFactory: itut_x780::ManagedObjectFactory
       {
              itut_x780::ManagedObject create
                     (in NameBindingType nameBinding,
                            // module name containing Name Binding info.
                     in MONameType superior,
                            // Name of containing object.
                     in string reqID,
                            // Requested ID value for name, will be
                            // empty if auto-naming is to be used.
                     out MONameType name,
                            // Entire name of newly created object.
                     in StringSetType packageNameList,
                            // List of packages requested.

                     in short historyRetention,
                     in AdministrativeStateType administrativeState,
                     in TimePeriodType granularityPeriod,
                     in MONameSetType thresholdDataInstanceList,
                     in short maxSuppressedIntervals,
                     in GeneralizedTimeType periodSynchronizationTime)
                     raises (itut_x780::ApplicationError,
                            itut_x780::CreateError);

       }; // interface CongDiscardCDFactory

/**
```

5.6.4 GenericTransportPmCD_F (Physical Transport Performance Monitoring Current

Data)

```
Retrieves attributes or current and history data counter values for counters
within the Physical Transport Protocol Monitoring grouping.

Additional inherited methods from Scanner provide for the setting of
AdministrativeState, retrieval of OperationalState, and setting of Granularity
Period.  Inherited methods from CurrentData provide for retrieval of the
SuspectFlag, retrieval of the ElapsedTime, activating or deactivating
HistoryRetention, associating threshold data with a current data instance,
and activating or deactivating the suppression of counters having all-zero
counts.

Applicable methods from Scanner include: administrativeStateGet,
administrativeStateSet, operationalStateGet, granularityPeriodGet, and
granularityPeriodSet.

Applicable methods from CurrentData include: suspectIntervalFlagGet,
suspectIntervalFlagDefaultSet, elapsedTimeGet, historyRetentionGet,
historyRetentionSet, thresholdDataInstanceListGet, thresholdDataInstanceListSet,
getMostRecent, getBetween, numSuppressedIntervalsGet, maxSuppressedIntervalsGet,
maxSuppressedIntervalsSet.

This object should support the following notifications: objectCreation,
objectDeletion, attributeValueChange, and stateChange.

A Threshold Crossing Alert (qualityOfServiceAlarm) is used to notify the
management system when any of the performance parameters exceeds a pre-set
threshold described in the associated ThresholdData object.

A containment relationship exists between an AtmLinkEndPhy or GenericTransportTTP and
its associated GenericTransportPmCD object.
*/

valuetype GenericTransportPmCDValueType: itut_q822d1::CurrentDataValueType
      {

/**
The codingViolationsPath parameter is used as a count of certain error events
occurring in the accumulation period. Sample error events include Frame
Synchronization and CRC errors for DS1 links, or P-bit and CP-bit parity
errors for DS3 links.
*/
            public unsigned long codingViolationsPath;
                  // GET
/**
The erroredSecondsPath parameter is a count of 1-second intervals containing
path errors. Sample path errors include CRC-6 errors (DS1), Severely-Errored
Frame defects (DS1/DS3), and P-bit Parity errors (DS3).
*/
            public unsigned long erroredSecondsPath;
                  // GET
/**
The severelyErroredSecondspath parameter is a count of 1-second intervals
with X or more path eooro events, or one or more Loss of Signal defect.
*/
            public unsigned long severelyErroredSecondsPath;
                  // GET
/**
The unavailableSecondsPath parameter is a count of 1-second intervals during
```

```
which the path is unavailable.
*/
            public unsigned long unavailableSecondsPath;
                  // GET
            public unsigned long failureCounterPath;
                  // GET
            public unsigned long farEndCodingViolationsPath;
                  // GET
            public unsigned long farEndErroredSecondsPath;
                  // GET
            public unsigned long farEndSeverelyErroredSecondsPath;
                  // GET
            public unsigned long farEndUnavailableSecondsPath;
                  // GET
            public unsigned long farEndFailureCounterPath;
                  // GET
/**
The codingViolationsLine parameter is count of certain error events occurring
in the accumulation period. Sample error events include Bipolar Violations
(BPVs) and Excessive Zeros (EXZs) occurring over a DS1/Ds3 link.
*/
            public unsigned long codingViolationsLine;
                  // GET
/**
A erroredSecondsLine is a count of 1-second intervals in which one or more
Line Coding Violation error events were detected.
*/
            public unsigned long erroredSecondsLine;
                  // GET

/**
severelyErroredSecondsLine is a count of 1-second intervals with X or more
BPVs plus EXZs, or one or more Loss of Signal defect.
*/
            public unsigned long severelyErroredSecondsLine;
                  // GET
/**
The lossOfSignalLine parameter is a count of 1-second intervals containing
one or more Loss of Signal defects.
*/
            public unsigned long lossOfSignalLine;
                  // GET
            public unsigned long failureCounterLine;
                  // GET
            public unsigned long farEndCodingViolationsLine;
                  // GET
            public unsigned long farEndErroredSecondsLine;
                  // GET
            public unsigned long farEndSeverelyErroredSecondsLine;
                  // GET
            public unsigned long farEndlossOfSignalLine;
                  // GET
            public unsigned long farEndFailureCounterLine;
                  // GET

            public unsigned long codingViolationsSection;
                  // GET
```

```
                  // Conditional, present if phy supports sections

            public unsigned long erroredSecondsSection;
                  // GET
                  // Conditional, present if phy supports sections

            public unsigned long severelyErroredSecondsSection;
                  // GET
                  // Conditional, present if phy supports sections

            public unsigned long lossOfSignalSection;
                  // GET
                  // Conditional, present if phy supports sections

            public unsigned long failureCounterSection;
                  // GET
                  // Conditional, present if phy supports sections

            public unsigned long farEndCodingViolationsSection;
                  // GET
                  // Conditional, present if phy supports sections

            public unsigned long farEndErroredSecondsSection;
                  // GET
                  // Conditional, present if phy supports sections

            public unsigned long farEndSeverelyErroredSecondsSection;
                  // GET
                  // Conditional, present if phy supports sections

            public unsigned long farEndlossOfSignalSection;
                  // GET
                  // Conditional, present if phy supports sections

            public unsigned long farEndFailureCounterSection;
                  // GET
                  // Conditional, present if phy supports sections

      }; // valuetype GenericTransportPmCDValueType



valuetype GenericTransportTTPHDValueType: itut_q822d1::HistoryDataValueType
      {

/**
The codingViolationsPath parameter is used as a count of certain error events
occurring in the accumulation period. Sample error events include Frame
Synchronization and CRC errors for DS1 links, or P-bit and CP-bit parity
errors for DS3 links.
*/
            public unsigned long codingViolationsPath;
                  // GET
/**
The erroredSecondsPath parameter is a count of 1-second intervals containing
path errors. Sample path errors include CRC-6 errors (DS1), Severely-Errored
Frame defects (DS1/DS3), and P-bit Parity errors (DS3).
```

```
*/
            public unsigned long erroredSecondsPath;
                  // GET
/**
The severelyErroredSecondspath parameter is a count of 1-second intervals
with X or more path eooro events, or one or more Loss of Signal defect.
*/
            public unsigned long severelyErroredSecondsPath;
                  // GET
/**
The unavailableSecondsPath parameter is a count of 1-second intervals during
which the path is unavailable.
*/
            public unsigned long unavailableSecondsPath;
                  // GET
            public unsigned long failureCounterPath;
                  // GET
            public unsigned long farEndCodingViolationsPath;
                  // GET
            public unsigned long farEndErroredSecondsPath;
                  // GET
            public unsigned long farEndSeverelyErroredSecondsPath;
                  // GET
            public unsigned long farEndUnavailableSecondsPath;
                  // GET
            public unsigned long farEndFailureCounterPath;
                  // GET
/**
The codingViolationsLine parameter is count of certain error events occurring
in the accumulation period. Sample error events include Bipolar Violations
(BPVs) and Excessive Zeros (EXZs) occurring over a DS1/Ds3 link.
*/
            public unsigned long codingViolationsLine;
                  // GET
/**
A erroredSecondsLine is a count of 1-second intervals in which one or more
Line Coding Violation error events were detected.
*/
            public unsigned long erroredSecondsLine;
                  // GET

/**
severelyErroredSecondsLine is a count of 1-second intervals with X or more
BPVs plus EXZs, or one or more Loss of Signal defect.
*/
            public unsigned long severelyErroredSecondsLine;
                  // GET
/**
The lossOfSignalLine parameter is a count of 1-second intervals containing
one or more Loss of Signal defects.
*/
            public unsigned long lossOfSignalLine;
                  // GET
            public unsigned long failureCounterLine;
                  // GET
            public unsigned long farEndCodingViolationsLine;
                  // GET
```

```
        public unsigned long farEndErroredSecondsLine;
              // GET
        public unsigned long farEndSeverelyErroredSecondsLine;
              // GET
        public unsigned long farEndlossOfSignalLine;
              // GET
        public unsigned long farEndFailureCounterLine;
              // GET

        public unsigned long codingViolationsSection;
              // GET
              // Conditional, present if phy supports sections

        public unsigned long erroredSecondsSection;
              // GET
              // Conditional, present if phy supports sections

        public unsigned long severelyErroredSecondsSection;
              // GET
              // Conditional, present if phy supports sections

        public unsigned long lossOfSignalSection;
              // GET
              // Conditional, present if phy supports sections

        public unsigned long failureCounterSection;
              // GET
              // Conditional, present if phy supports sections

        public unsigned long farEndCodingViolationsSection;
              // GET
              // Conditional, present if phy supports sections

        public unsigned long farEndErroredSecondsSection;
              // GET
              // Conditional, present if phy supports sections

        public unsigned long farEndSeverelyErroredSecondsSection;
              // GET
              // Conditional, present if phy supports sections

        public unsigned long farEndlossOfSignalSection;
              // GET
              // Conditional, present if phy supports sections

        public unsigned long farEndFailureCounterSection;
              // GET
              // Conditional, present if phy supports sections


    }; // valuetype GenericTransportTTPHDValueType

    interface GenericTransportPmCD_F: itut_q822d1::CurrentData_F
    {

         unsigned long codingViolationsPathGet
              (in MONameType name)
```

```
        raises (itut_x780::ApplicationError);

    unsigned long erroredSecondsPathGet
        (in MONameType name)
        raises (itut_x780::ApplicationError);

    unsigned long severelyErroredSecondsPathGet
        (in MONameType name)
        raises (itut_x780::ApplicationError);

    unsigned long unavailableSecondsPathGet
        (in MONameType name)
        raises (itut_x780::ApplicationError);

    unsigned long failureCounterPathGet
        (in MONameType name)
        raises (itut_x780::ApplicationError);

    unsigned long farEndCodingViolationsPathGet
        (in MONameType name)
        raises (itut_x780::ApplicationError);

    unsigned long farEndErroredSecondsPathGet
        (in MONameType name)
        raises (itut_x780::ApplicationError);

    unsigned long farEndSeverelyErroredSecondsPathGet
        (in MONameType name)
        raises (itut_x780::ApplicationError);

    unsigned long farEndUnavailableSecondsPathGet
        (in MONameType name)
        raises (itut_x780::ApplicationError);

    unsigned long farEndFailureCounterPathGet
        (in MONameType name)
        raises (itut_x780::ApplicationError);

    unsigned long codingViolationsLineGet
        (in MONameType name)
        raises (itut_x780::ApplicationError);

    unsigned long erroredSecondsLineGet
        (in MONameType name)
        raises (itut_x780::ApplicationError);

    unsigned long severelyErroredSecondsLineGet
        (in MONameType name)
        raises (itut_x780::ApplicationError);

    unsigned long lossOfSignalLineGet
        (in MONameType name)
        raises (itut_x780::ApplicationError);

    unsigned long failureCounterLineGet
        (in MONameType name)
        raises (itut_x780::ApplicationError);
```

```
unsigned long farEndCodingViolationsLineGet
     (in MONameType name)
     raises (itut_x780::ApplicationError);

unsigned long farEndErroredSecondsLineGet
     (in MONameType name)
     raises (itut_x780::ApplicationError);

unsigned long farEndSeverelyErroredSecondsLineGet
     (in MONameType name)
     raises (itut_x780::ApplicationError);

unsigned long farEndlossOfSignalLineGet
     (in MONameType name)
     raises (itut_x780::ApplicationError);

unsigned long farEndFailureCounterLineGet
     (in MONameType name)
     raises (itut_x780::ApplicationError);


unsigned long codingViolationsSectionGet
     (in MONameType name)
     raises (itut_x780::ApplicationError);

unsigned long erroredSecondsSectionGet
     (in MONameType name)
     raises (itut_x780::ApplicationError);

unsigned long severelyErroredSecondsSectionGet
     (in MONameType name)
     raises (itut_x780::ApplicationError);

unsigned long lossOfSignalSectionGet
     (in MONameType name)
     raises (itut_x780::ApplicationError);

unsigned long failureCounterSectionGet
     (in MONameType name)
     raises (itut_x780::ApplicationError);

unsigned long farEndCodingViolationsSectionGet
     (in MONameType name)
     raises (itut_x780::ApplicationError);

unsigned long farEndErroredSecondsSectionGet
     (in MONameType name)
     raises (itut_x780::ApplicationError);

unsigned long farEndSeverelyErroredSecondsSectionGet
     (in MONameType name)
     raises (itut_x780::ApplicationError);

unsigned long farEndlossOfSignalSectionGet
     (in MONameType name)
     raises (itut_x780::ApplicationError);
```

```
            unsigned long farEndFailureCounterSectionGet
                  (in MONameType name)
                  raises (itut_x780::ApplicationError);

      }; // interface GenericTransportPmCD_F


      interface GenericTransportPmCDFactory: itut_x780::ManagedObjectFactory
      {
            itut_x780::ManagedObject create
                  (in NameBindingType nameBinding,
                        // module name containing Name Binding info.
                  in MONameType superior,
                        // Name of containing object.
                  in string reqID,
                        // Requested ID value for name, will be
                        // empty if auto-naming is to be used.
                  out MONameType name,
                        // Entire name of newly created object.
                  in StringSetType packageNameList,
                        // List of packages requested.

                  in short historyRetention,
                  in AdministrativeStateType administrativeState,
                  in TimePeriodType granularityPeriod,
                  in MONameSetType thresholdDataInstanceList,
                  in short maxSuppressedIntervals,
                  in GeneralizedTimeType periodSynchronizationTime)
                  raises (itut_x780::ApplicationError,
                        itut_x780::CreateError);

      }; // interface GenericTransportPmCDFactory


/**
```

5.6.5 PmOamCD_F (PM OAM Cell Monitoring Current Data)

Retrieves attributes or current and history data counter values for counters within the ATM PM OAM Cell Monitoring grouping, namely, numberPmOamLostCells, numberPmOamMisinsertedCells, numberPmOamUserCells, numberPmOamFarEndLostCells, numberPmOamFarEndMisinsertedCells and numberPmOamFarEndUserCells.

Additional inherited methods from Scanner provide for the setting of AdministrativeState, retrieval of OperationalState, and setting of Granularity Period. Inherited methods from CurrentData provide for retrieval of the SuspectFlag, retrieval of the ElapsedTime, activating or deactivating HistoryRetention, associating threshold data with a current data instance, and activating or deactivating the suppression of counters having all-zero counts.

Applicable methods from Scanner include: administrativeStateGet, administrativeStateSet, operationalStateGet, granularityPeriodGet, and granularityPeriodSet.

Applicable methods from CurrentData include: suspectIntervalFlagGet, suspectIntervalFlagDefaultSet, elapsedTimeGet, historyRetentionGet, historyRetentionSet, thresholdDataInstanceListGet, thresholdDataInstanceListSet, getMostRecent, getBetween, numSuppressedIntervalsGet, maxSuppressedIntervalsGet, maxSuppressedIntervalsSet.

```
This object should support the following notifications: objectCreation,
objectDeletion, attributeValueChange, and stateChange.

A Threshold Crossing Alert (qualityOfServiceAlarm) is used to notify the
management system when any of the performance parameters exceeds a pre-set
threshold described in the associated ThresholdData object.

A containment relationship exists between an AtmLinkEnd and its associated
PmOamCurrentData object.
*/
       valuetype PmOamCDValueType: itut_q822d1::CurrentDataValueType
       {
              public unsigned long numberPmOamLostCells;
                     // GET
              public unsigned long numberPmOamMisinsertedCells;
                     // GET
              public unsigned long numberPmOamUserCells;
                     // GET
              public unsigned long numberPmOamFarEndLostCells;
                     // GET
              public unsigned long numberPmOamFarEndMisinsertedCells;
                     // GET
              public unsigned long numberPmOamFarEndUserCells;
                     // GET
       }; // valuetype PmOamCDValueType


       valuetype PmOamHDValueType: itut_q822d1::HistoryDataValueType
       {
              public unsigned long numberPmOamLostCells;
                     // GET
              public unsigned long numberPmOamMisinsertedCells;
                     // GET
              public unsigned long numberPmOamUserCells;
                     // GET
              public unsigned long numberPmOamFarEndLostCells;
                     // GET
              public unsigned long numberPmOamFarEndMisinsertedCells;
                     // GET
              public unsigned long numberPmOamFarEndUserCells;
                     // GET
       }; // valuetype PmOamHDValueType


       interface PmOamCD_F: itut_q822d1::CurrentData_F
       {
/**
Number of lost cells measured by PM OAM.
*/
              unsigned long numberPmOamLostCellsGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);
/**
Number of misinserted cells measured by PM OAM.
*/
              unsigned long numberPmOamMisinsertedCellsGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);

/**
Number of user cells measured by PM OAM.
*/
              unsigned long numberPmOamUserCellsGet
```

```
                    (in MONameType name)
                    raises (itut_x780::ApplicationError);
/**
Number of Far-End Lost cells measured by PM OAM.
*/
              unsigned long numberPmOamFarEndLostCellsGet
                    (in MONameType name)
                    raises (itut_x780::ApplicationError);
/**
Number of Far End Misinserted cells measured by PM OAM.
*/
              unsigned long numberPmOamFarEndMisinsertedCellsGet
                    (in MONameType name)
                    raises (itut_x780::ApplicationError);
/**
Number of Far End User cells measured by PM OAM.
*/
              unsigned long numberPmOamFarEndUserCellsGet
                    (in MONameType name)
                    raises (itut_x780::ApplicationError);

       }; // interface PmOamCD_F


       interface PmOamCDFactory: itut_x780::ManagedObjectFactory
       {
              itut_x780::ManagedObject create
                    (in NameBindingType nameBinding,
                          // module name containing Name Binding info.
                    in MONameType superior,
                          // Name of containing object.
                    in string reqID,
                          // Requested ID value for name, will be
                          // empty if auto-naming is to be used.
                    out MONameType name,
                          // Entire name of newly created object.
                    in StringSetType packageNameList,
                          // List of packages requested.

                    in short historyRetention,
                    in AdministrativeStateType administrativeState,
                    in TimePeriodType granularityPeriod,
                    in MONameSetType thresholdDataInstanceList,
                    in short maxSuppressedIntervals,
                    in GeneralizedTimeType periodSynchronizationTime)
                    raises (itut_x780::ApplicationError,
                          itut_x780::CreateError);

       }; // interface PmOamCDFactory



/**
```

5.6.6 TcAdaptProtMonCD_F (TC Adaptor Protocol Monitoring Current Data)

```
Retrieves attributes or current and history data counter values for counters
within the TC Adaptor Protocol Monitoring grouping, namely,
numberDiscCellsHECViolat.

Additional inherited methods from Scanner provide for the setting of
```

```
AdministrativeState, retrieval of OperationalState, and setting of Granularity
Period.  Inherited methods from CurrentData provide for retrieval of the
SuspectFlag, retrieval of the ElapsedTime, activating or deactivating
HistoryRetention, associating threshold data with a current data instance, and
activating or deactivating the suppression of counters having all-zero counts.

Applicable methods from Scanner include: administrativeStateGet,
administrativeStateSet, operationalStateGet, granularityPeriodGet, and
granularityPeriodSet.

Applicable methods from CurrentData include: suspectIntervalFlagGet,
suspectIntervalFlagDefaultSet, elapsedTimeGet, historyRetentionGet,
historyRetentionSet, thresholdDataInstanceListGet, thresholdDataInstanceListSet,
getMostRecent, getBetween, numSuppressedIntervalsGet, maxSuppressedIntervalsGet,
maxSuppressedIntervalsSet.

This object should support the following notifications: objectCreation,
objectDeletion, attributeValueChange, and stateChange.

A Threshold Crossing Alert (qualityOfServiceAlarm) is used to notify the
management system when any of the performance parameters exceeds a pre-set
threshold described in the associated ThresholdData object.

A containment relationship exists between an AtmLinkEnd and its associated
TcAdaptProtMonCurrentData object.
*/
       valuetype TcAdaptProtMonCDValueType: itut_q822d1::CurrentDataValueType
       {
              public unsigned long numberDiscCellsHECViolat;
                     // GET
       }; // valuetype TcAdaptProtMonCDValueType


       valuetype TcAdaptProtMonHDValueType: itut_q822d1::HistoryDataValueType
       {
              public unsigned long numberDiscCellsHECViolat;
                     // GET
       }; // valuetype TcAdaptProtMonHDValueType


       interface TcAdaptProtMonCD_F: itut_q822d1::CurrentData_F
       {
/**
Count of cells discarded due to HEC Violations; thresholded count.
*/
              unsigned long numberDiscCellsHECViolatGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);
       }; // interface TcAdaptProtMonCD_F


       interface TcAdaptProtMonCDFactory: itut_x780::ManagedObjectFactory
       {
              itut_x780::ManagedObject create
                     (in NameBindingType nameBinding,
                            // module name containing Name Binding info.
                     in MONameType superior,
                            // Name of containing object.
                     in string reqID,
                            // Requested ID value for name, will be
                            // empty if auto-naming is to be used.
                     out MONameType name,
                            // Entire name of newly created object.
```

```
                    in StringSetType packageNameList,
                          // List of packages requested.

                    in short historyRetention,
                    in AdministrativeStateType administrativeState,
                    in TimePeriodType granularityPeriod,
                    in MONameSetType thresholdDataInstanceList,
                    in short maxSuppressedIntervals,
                    in GeneralizedTimeType periodSynchronizationTime)
                    raises (itut_x780::ApplicationError,
                          itut_x780::CreateError);

      }; // interface TcAdaptProtMonCDFactory



/**
```

5.6.7 TrafficLoadCD_F (Traffic Load Current Data)

```
Retrieves attributes or current and history data counter values for counters
within the ATM Traffic Load grouping, namely, numberCellsRecvd and
numberCellsTrnsd.

Additional inherited methods from Scanner provide for the setting of
AdministrativeState, retrieval of OperationalState, and setting of Granularity
Period.  Inherited methods from CurrentData provide for retrieval of the
SuspectFlag, retrieval of the ElapsedTime, activating or deactivating
HistoryRetention, associating threshold data with a current data instance, and
activating or deactivating the suppression of counters having all-zero counts.

Applicable methods from Scanner include: administrativeStateGet,
administrativeStateSet, operationalStateGet, granularityPeriodGet, and
granularityPeriodSet.

Applicable methods from CurrentData include: suspectIntervalFlagGet,
suspectIntervalFlagDefaultSet, elapsedTimeGet, historyRetentionGet,
historyRetentionSet, thresholdDataInstanceListGet, thresholdDataInstanceListSet,
getMostRecent, getBetween, numSuppressedIntervalsGet, maxSuppressedIntervalsGet,
maxSuppressedIntervalsSet.

This object should support the following notifications: objectCreation,
objectDeletion, attributeValueChange, and stateChange.

Threshold Crossing Alert (qualityOfServiceAlarm) is not supported by the
AtmTrafficLoadCurrentData object.

A containment relationship exists between an AtmLinkEnd or AtmNetworkCTP and
the associated AtmTrafficLoadCurrentData object.
*/
      valuetype TrafficLoadCDValueType: itut_q822d1::CurrentDataValueType
      {
            public unsigned long numberCellsRecvd;
                  // GET
            public unsigned long numberCellsTrnsd;
                  // GET
      }; // valuetype TrafficLoadCDValueType


      valuetype TrafficLoadHDValueType: itut_q822d1::HistoryDataValueType
      {
```

```
              public unsigned long numberCellsRecvd;
                     // GET
              public unsigned long numberCellsTrnsd;
                     // GET
       }; // valuetype TrafficLoadHDValueType


       interface TrafficLoadCD_F: itut_q822d1::CurrentData_F
       {
/**
Number of cells received.
*/
              unsigned long numberCellsRecvdGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);

/**
Number of cells transmitted.
*/
              unsigned long numberCellsTrnsdGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);

       }; // interface TrafficLoadCD_F


       interface TrafficLoadCDFactory: itut_x780::ManagedObjectFactory
       {
              itut_x780::ManagedObject create
                     (in NameBindingType nameBinding,
                            // module name containing Name Binding info.
                     in MONameType superior,
                            // Name of containing object.
                     in string reqID,
                            // Requested ID value for name, will be
                            // empty if auto-naming is to be used.
                     out MONameType name,
                            // Entire name of newly created object.
                     in StringSetType packages,
                            // List of packages requested.

                     in short historyRetention,
                     in AdministrativeStateType administrativeState,
                     in TimePeriodType granularityPeriod,
                     in MONameSetType thresholdDataInstanceList,
                     in short maxSuppressedIntervals,
                     in GeneralizedTimeType periodSynchronizationTime)
                     raises (itut_x780::ApplicationError,
                            itut_x780::CreateError);

       }; // interface TrafficLoadCDFactory




/**
```

5.6.8 UpcNpcDisagreementsCD_F (UPC and NPC Disagreements Current Data)

```
Retrieves attributes or current and history data counter values for counters
within the ATM UPC and NPC Disagreements grouping, namely, numberCellsDiscCong,
numberSuccessfullyPassedUpcNpcCells, numberDiscardCLP0UpcNpcCells, and
```

```
numberSuccessfullyPassedCLP0UpcNpcCells.

Additional inherited methods from Scanner provide for the setting of
AdministrativeState, retrieval of OperationalState, and setting of Granularity
Period.  Inherited methods from CurrentData provide for retrieval of the
SuspectFlag, retrieval of the ElapsedTime, activating or deactivating
HistoryRetention, associating threshold data with a current data instance, and
activating or deactivating the suppression of counters having all-zero counts.

Applicable methods from Scanner include: administrativeStateGet,
administrativeStateSet, operationalStateGet, granularityPeriodGet, and
granularityPeriodSet.

Applicable methods from CurrentData include: suspectIntervalFlagGet,
suspectIntervalFlagDefaultSet, elapsedTimeGet, historyRetentionGet,
historyRetentionSet, thresholdDataInstanceListGet,
thresholdDataInstanceListSet,getMostRecent, getBetween, numSuppressedIntervalsGet,
maxSuppressedIntervalsGet,
maxSuppressedIntervalsSet.

This object should support the following notifications: objectCreation,
objectDeletion, attributeValueChange, and stateChange.

A Threshold Crossing Alert (qualityOfServiceAlarm) is used to notify the
management system when any of the performance parameters exceeds a pre-set
threshold described in the associated ThresholdData object.

A containment relationship exists between an AtmLinkEnd and its associated
UpcNpcDisagreementsCurrentData object.
*/
       valuetype UpcNpcDisagreementsCDValueType:
              itut_q822d1::CurrentDataValueType
       {
              public unsigned long numberDiscardUpcNpcCells;
                     // GET
              public unsigned long numberSuccessfullyPassedUpcNpcCells;
                     // GET
              public unsigned long numberDiscardCLP0UpcNpcCells;
                     // GET
              public unsigned long numberSuccessfullyPassedCLP0UpcNpcCells;
                     // GET
       }; // valuetype UpcNpcDisagreementsCDValueType


       valuetype UpcNpcDisagreementsHDValueType:
              itut_q822d1::HistoryDataValueType
       {
              public unsigned long numberDiscardUpcNpcCells;
                     // GET
              public unsigned long numberSuccessfullyPassedUpcNpcCells;
                     // GET
              public unsigned long numberDiscardCLP0UpcNpcCells;
                     // GET
              public unsigned long numberSuccessfullyPassedCLP0UpcNpcCells;
                     // GET
       }; // valuetype UpcNpcDisagreementsHDValueType


       interface UpcNpcDisagreementsCD_F: itut_q822d1::CurrentData_F
       {
/**
Number of cells discarded due to policing (UPC/NPC); thresholded count.
*/
```

```
              unsigned long numberDiscardUpcNpcCellsGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);
/**
Number of cells marked by UPC/NPC that are passed.
*/
              unsigned long numberSuccessfullyPassedUpcNpcCellsGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);
/**
Number of CLP=0 cells discarded due to policing; thresholded count.
*/
              unsigned long numberDiscardCLP0UpcNpcCellsGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);
/**
Number of CLP=0 cells marked by UPC/NPC that are passed.
*/
              unsigned long numberSuccessfullyPassedCLP0UpcNpcCellsGet
                     (in MONameType name)
                     raises (itut_x780::ApplicationError);

       }; // interface UpcNpcDisagreementsCD_F


       interface UpcNpcDisagreementsCDFactory: itut_x780::ManagedObjectFactory
       {
              itut_x780::ManagedObject create
                     (in NameBindingType nameBinding,
                            // module name containing Name Binding info.
                     in MONameType superior,
                            // Name of containing object.
                     in string reqID,
                            // Requested ID value for name, will be
                            // empty if auto-naming is to be used.
                     out MONameType name,
                            // Entire name of newly created object.
                     in StringSetType packageNameList,
                            // List of packages requested.

                     in short historyRetention,
                     in AdministrativeStateType administrativeState,
                     in TimePeriodType granularityPeriod,
                     in MONameSetType thresholdDataInstanceList,
                     in short maxSuppressedIntervals,
                     in GeneralizedTimeType periodSynchronizationTime)
                     raises (itut_x780::ApplicationError,
                            itut_x780::CreateError);

       }; // interface UpcNpcDisagreementsCDFactory
```

```
/**
```

5.7. *Performance Management – Fine-grained*

```
This section defines the interfaces for supporting Performance Management (PM)
for ATM. Further discussion of PM-related managed entities and attributes is
provided in Sections 3.17, 3.18, 3.26, 3.26. 3.41, 3.42, 3.90, 3.91 and 3.92
of the ATM Network View Requirements and Logical MIB.  All of the performance
management aspects for module atmf_m4nw_v2 are grouped into this section.
It draws upon the technology independent performance management module in
ITU-T Rec. Q.822.1 and adds extensions that are specific to ATM.


The behaviour of the fine-grained interfaces are identical to the corresponding
facade interfaces. Therefore, comments are not included in the fine-grained
interfaces. Readers are referred to the facade interface in Section 5.6
for the object behaviour, value types, factories of the facade interface.

This section can be omitted from IDL if a management system only supports
fine-grained interface.
*/


/**
Interfaces imported from itut_q822d1:

CurrentData
*/

/*
```

5.7.1 Forward Declarations

```
*/




/**
```

5.7.2 CellProtocolMonCD (Cell Protocol Monitoring Current Data)

```
*/


      interface CellProtocolMonCD: itut_q822d1::CurrentData
      {
            unsigned long numberDiscCellsProtErrGet()
                  raises (itut_x780::ApplicationError);
            unsigned long numberRecvOAMCellsGet()
                  raises (itut_x780::ApplicationError);

      }; // interface CellProtocolMonCD




/**
```

5.7.3 CongDiscardCD (ATM Congestion Discards Current Data)

```
*/

      interface CongDiscardCD: itut_q822d1::CurrentData
      {
            unsigned long numberCellsDiscCongGet()
                  raises (itut_x780::ApplicationError);

            unsigned long numberCLP0CellsDiscGet()
                  raises (itut_x780::ApplicationError);
      }; // interface CongDiscardCD



/**
```

5.7.4 GenericTransportPmCD (Physical Transport Performance Monitoring Current Data)

```
*/


      interface GenericTransportPmCD: itut_q822d1::CurrentData
      {

             unsigned long codingViolationsPathGet()
                  raises (itut_x780::ApplicationError);

             unsigned long erroredSecondsPathGet()
                  raises (itut_x780::ApplicationError);

             unsigned long severelyErroredSecondsPathGet()
                  raises (itut_x780::ApplicationError);

             unsigned long unavailableSecondsPathGet()
                  raises (itut_x780::ApplicationError);

             unsigned long failureCounterPathGet()
                  raises (itut_x780::ApplicationError);

             unsigned long farEndCodingViolationsPathGet()
                  raises (itut_x780::ApplicationError);

             unsigned long farEndErroredSecondsPathGet()
                  raises (itut_x780::ApplicationError);

             unsigned long farEndSeverelyErroredSecondsPathGet()
                  raises (itut_x780::ApplicationError);

             unsigned long farEndUnavailableSecondsPathGet()
                  raises (itut_x780::ApplicationError);

             unsigned long farEndFailureCounterPathGet()
                  raises (itut_x780::ApplicationError);

             unsigned long codingViolationsLineGet()
                  raises (itut_x780::ApplicationError);
```

```
unsigned long erroredSecondsLineGet()
     raises (itut_x780::ApplicationError);

unsigned long severelyErroredSecondsLineGet()
     raises (itut_x780::ApplicationError);

unsigned long lossOfSignalLineGet()
     raises (itut_x780::ApplicationError);

unsigned long failureCounterLineGet()
     raises (itut_x780::ApplicationError);

unsigned long farEndCodingViolationsLineGet()
     raises (itut_x780::ApplicationError);

unsigned long farEndErroredSecondsLineGet()
     raises (itut_x780::ApplicationError);

unsigned long farEndSeverelyErroredSecondsLineGet()
     raises (itut_x780::ApplicationError);

unsigned long farEndlossOfSignalLineGet()
     raises (itut_x780::ApplicationError);

unsigned long farEndFailureCounterLineGet()
     raises (itut_x780::ApplicationError);

unsigned long codingViolationsSectionGet()
     raises (itut_x780::ApplicationError);

unsigned long erroredSecondsSectionGet()
     raises (itut_x780::ApplicationError);

unsigned long severelyErroredSecondsSectionGet()
     raises (itut_x780::ApplicationError);

unsigned long lossOfSignalSectionGet()
     raises (itut_x780::ApplicationError);

unsigned long failureCounterSectionGet()
     raises (itut_x780::ApplicationError);

unsigned long farEndCodingViolationsSectionGet()
     raises (itut_x780::ApplicationError);

unsigned long farEndErroredSecondsSectionGet()
     raises (itut_x780::ApplicationError);

unsigned long farEndSeverelyErroredSecondsSectionGet()
     raises (itut_x780::ApplicationError);

unsigned long farEndlossOfSignalSectionGet()
     raises (itut_x780::ApplicationError);

unsigned long farEndFailureCounterSectionGet()
     raises (itut_x780::ApplicationError);
```

```
	}; // interface GenericTransportPmCD



/**
```

5.7.5 PmOamCD (PM OAM Cell Monitoring Current Data)

```
*/


	interface PmOamCD: itut_q822d1::CurrentData
	{
		unsigned long numberPmOamLostCellsGet()
			raises (itut_x780::ApplicationError);

		unsigned long numberPmOamMisinsertedCellsGet()
			raises (itut_x780::ApplicationError);

		unsigned long numberPmOamUserCellsGet()
			raises (itut_x780::ApplicationError);

		unsigned long numberPmOamFarEndLostCellsGet()
			raises (itut_x780::ApplicationError);

		unsigned long numberPmOamFarEndMisinsertedCellsGet()
			raises (itut_x780::ApplicationError);

		unsigned long numberPmOamFarEndUserCellsGet()
			raises (itut_x780::ApplicationError);

	}; // interface PmOamCD




/**
```

5.7.6 TcAdaptProtMonCD (TC Adaptor Protocol Monitoring Current Data)

```
*/


	interface TcAdaptProtMonCD: itut_q822d1::CurrentData
	{
		unsigned long numberDiscCellsHECViolatGet()
			raises (itut_x780::ApplicationError);
	}; // interface TcAdaptProtMonCD




/**
```

5.7.7 TrafficLoadCD (Traffic Load Current Data)

```
*/
```

```
	interface TrafficLoadCD: itut_q822d1::CurrentData
	{
		unsigned long numberCellsRecvdGet()
			raises (itut_x780::ApplicationError);

		unsigned long numberCellsTrnsdGet()
			raises (itut_x780::ApplicationError);

	}; // interface TrafficLoadCD




/**
```

5.7.8 UpcNpcDisagreementsCD (UPC and NPC Disagreements Current Data)

```
*/


	interface UpcNpcDisagreementsCD: itut_q822d1::CurrentData
	{
		unsigned long numberDiscardUpcNpcCellsGet()
			raises (itut_x780::ApplicationError);

		unsigned long numberSuccessfullyPassedUpcNpcCellsGet()
			raises (itut_x780::ApplicationError);

		unsigned long numberDiscardCLP0UpcNpcCellsGet()
			raises (itut_x780::ApplicationError);

		unsigned long numberSuccessfullyPassedCLP0UpcNpcCellsGet()
			raises (itut_x780::ApplicationError);

	}; // interface UpcNpcDisagreementsCD
```

```
/**
```

5.8. *Name Binding*

```
*/
       module NameBinding
       {

       typedef itut_x780::DeletePolicyType DeletePolicyType;

/**
```

5.8.1 AbstractAalProfile

```
*/


              module AbstractAalProfile_AtmSubnetwork
              {
                     const string  superiorClass =
                            "atmf_m4nw_v2::AtmSubnetwork";
                     const booleansuperiorSubclassesAllowed = TRUE;
                     const string  subordinateClass =
                            "atmf_m4nw_v2::AbstractAalProfile";
                     const booleansubordinateSubclassesAllowed = TRUE;
                     const booleanmanagerCreatesAllowed = TRUE;
                     const DeletePolicyType deletePolicy =
                            itut_x780::deleteOnlyIfNoContainedObjects;
                     const string  kind = " AbstractAalProfile";
              }; // module AbstractAalProfile_AtmSubnetwork


/**
```

5.8.2 AtmImaGroupTP

```
*/
              module AtmImaGroupTP_ManagedElement
              {
                     const string  superiorClass =
                            " itut_m3120::ManagedElement";
                     const booleansuperiorSubclassesAllowed = TRUE;
                     const string  subordinateClass =
                            "atmf_m4nw_v2::AtmImaGroupTP";
                     const booleansubordinateSubclassesAllowed = TRUE;
                     const booleanmanagerCreatesAllowed = TRUE;
                     const DeletePolicyType deletePolicy =
                            itut_x780::deleteOnlyIfNoContainedObjects;
                     const string  kind = "AtmImaGroupTP";
              }; // module AtmImaGroupTP_ManagedElement

/**
```

5.8.3 AtmImaLinkTP

```
*/
              module AtmImaLinkTP_ManagedElement
              {
                     const string  superiorClass =
                            " itut_m3120::ManagedElement";
                     const booleansuperiorSubclassesAllowed = TRUE;
                     const string  subordinateClass =
                            "atmf_m4nw_v2::AtmImaLinkTP";
```

```
                const booleansubordinateSubclassesAllowed = TRUE;
                const booleanmanagerCreatesAllowed = TRUE;
                const DeletePolicyType deletePolicy =
                       itut_x780::deleteOnlyIfNoContainedObjects;
                const string kind = "AtmImaLinkTP";
        }; // module AtmImaLinkTP_ManagedElement


/**
```

5.8.4 AtmLink

```
*/
        module AtmLink_LayerND
        {
                const string superiorClass =
                       "itut_m3120:: LayerND";
                const booleansuperiorSubclassesAllowed = TRUE;
                const string subordinateClass =
                       "atmf_m4nw_v2::AtmLink";
                const booleansubordinateSubclassesAllowed = TRUE;
                const booleanmanagerCreatesAllowed = TRUE;
                const DeletePolicyType deletePolicy =
                       itut_x780::deleteOnlyIfNoContainedObjects;
                const string kind = "AtmLink";
        }; // module AtmLink_LayerND



/**
```

5.8.5 AtmLinkEnd

```
*/
        module AtmLinkEnd_LayerND
        {
                const string superiorClass =
                       "itut_m3120::LayerND";
                const booleansuperiorSubclassesAllowed = TRUE;
                const string subordinateClass =
                       "atmf_m4nw_v2::AtmLinkEnd";
                const booleansubordinateSubclassesAllowed = TRUE;
                const booleanmanagerCreatesAllowed = TRUE;
                const DeletePolicyType deletePolicy =
                       itut_x780::deleteOnlyIfNoContainedObjects;
                const string kind = "AtmLinkEnd";
        }; // module AtmLinkEnd_LayerND


/**
```

5.8.6 AtmLinkEndPhy

```
*/
        module AtmLinkEndPhy_LayerND
        {
                const string superiorClass =
                       "itut_m3120::LayerND";
                const booleansuperiorSubclassesAllowed = TRUE;
                const string subordinateClass =
                       "atmf_m4nw_v2::AtmLinkEndPhy";
                const booleansubordinateSubclassesAllowed = TRUE;
                const booleanmanagerCreatesAllowed = FALSE;
```

```
                const DeletePolicyType deletePolicy =
                       itut_x780::notDeletable;
                const string  kind = "AtmLinkEndPhy";
        }; // module AtmLinkEndPhy_LayerND



/**
```

5.8.7 AtmNetworkAccessProfile

```
*/
        module AtmNetworkAccessProfile_AtmSubnetwork
        {
                const string  superiorClass =
                       "atmf_m4nw_v2::AtmSubnetwork";
                const booleansuperiorSubclassesAllowed = TRUE;
                const string  subordinateClass =
                       "atmf_m4nw_v2::AtmNetworkAccessProfile";
                const booleansubordinateSubclassesAllowed = TRUE;
                const booleanmanagerCreatesAllowed = TRUE;
                const DeletePolicyType deletePolicy =
                       itut_x780::deleteOnlyIfNoContainedObjects;
                const string  kind = "AtmNetworkAccessProfile";
        }; // module AtmNetworkAccessProfile_AtmSubnetwork



/**
```

5.8.8 AtmNetworkCTP

```
*/
        module AtmNetworkCTP_AtmLinkEnd
        {
                const string  superiorClass =
                       "atmf_m4nw_v2::AtmLinkEnd";
                const booleansuperiorSubclassesAllowed = TRUE;
                const string  subordinateClass =
                       "atmf_m4nw_v2::AtmNetworkCTP";
                const booleansubordinateSubclassesAllowed = TRUE;
                const booleanmanagerCreatesAllowed = TRUE;
                const DeletePolicyType deletePolicy =
                       itut_x780::deleteOnlyIfNoContainedObjects;
                const string  kind = "AtmNetworkCTP";
        }; // module AtmNetworkCTP_AtmLinkEnd



/**
```

5.8.9 AtmNetworkTTP

```
*/
        module AtmNetworkTTP_AtmLinkEnd
        {
                const string  superiorClass =
                       "atmf_m4nw_v2::AtmLinkEnd";
                const booleansuperiorSubclassesAllowed = TRUE;
                const string  subordinateClass =
                       "atmf_m4nw_v2::AtmNetworkTTP";
```

```
                const booleansubordinateSubclassesAllowed = TRUE;
                const booleanmanagerCreatesAllowed = TRUE;
                const DeletePolicyType deletePolicy =
                        itut_x780::deleteOnlyIfNoContainedObjects;
                const string kind = "AtmNetworkTTP";
        }; // module AtmNetworkTTP_AtmLinkEnd



/**
```

5.8.10 AtmRoutingProfile

```
*/
        module AtmRoutingProfile_AtmSubnetwork
        {
                const string superiorClass =
                        "atmf_m4nw_v2::AtmSubnetwork";
                const booleansuperiorSubclassesAllowed = TRUE;
                const string subordinateClass =
                        "atmf_m4nw_v2::AtmNetworkRoutingProfile";
                const booleansubordinateSubclassesAllowed = TRUE;
                const booleanmanagerCreatesAllowed = TRUE;
                const DeletePolicyType deletePolicy =
                        itut_x780::deleteOnlyIfNoContainedObjects;
                const string kind = "AtmNetworkRoutingProfile";
        }; // module AtmRoutingProfile_AtmSubnetwork



/**
```

5.8.11 AtmSNC

```
*/
        module AtmSNC_AtmSubnetwork
        {
                const string superiorClass =
                        "atmf_m4nw_v2::AtmSubnetwork";
                const booleansuperiorSubclassesAllowed = TRUE;
                const string subordinateClass =
                        "atmf_m4nw_v2::AtmNetworkSNC";
                const booleansubordinateSubclassesAllowed = TRUE;
                const booleanmanagerCreatesAllowed = TRUE;
                const DeletePolicyType deletePolicy =
                        itut_x780::notDeletable;
                const string kind = "AtmSNC";
        }; // module AtmSNC_AtmSubnetwork



/**
```

5.8.12 AtmSubnetwork

```
*/
        module AtmSubnetwork_LayerND
        {
                const string superiorClass =
                        "itut_m3120::LayerND";
                const booleansuperiorSubclassesAllowed = TRUE;
                const string subordinateClass =
                        "atmf_m4nw_v2::AtmSubnetwork";
```

```
                const booleansubordinateSubclassesAllowed = TRUE;
                const booleanmanagerCreatesAllowed = FALSE;
                const DeletePolicyType deletePolicy =
                       itut_x780::notDeletable;
                const string  kind = "AtmSubnetwork";
        }; // module AtmSubnetwork_LayerND



/**
```

5.8.13 AtmAbstractTrafficDesc

```
*/
        module AtmAbstractTrafficDesc_AtmSubnetwork
        {
                const string  superiorClass =
                       "atmf_m4nw_v2::AtmSubnetwork";
                const booleansuperiorSubclassesAllowed = TRUE;
                const string  subordinateClass =
                       "atmf_m4nw_v2::AtmNetworkAbstractTrafficDesc";
                const booleansubordinateSubclassesAllowed = TRUE;
                const booleanmanagerCreatesAllowed = TRUE;
                const DeletePolicyType deletePolicy =
                       itut_x780::deleteOnlyIfNoContainedObjects;
                const string  kind = "AtmAbstractTrafficDesc";
        }; // module AtmAbstractTrafficDesc_AtmSubnetwork

/**
```

5.8.14 CesServiceProfile

```
*/
        module CesServiceProfile_AtmSubnetwork
        {
                const string  superiorClass =
                       "itut_m3120::AtmSubnetwork";
                const booleansuperiorSubclassesAllowed = TRUE;
                const string  subordinateClass =
                       "CesServiceProfile";
                const booleansubordinateSubclassesAllowed = TRUE;
                const booleanmanagerCreatesAllowed = TRUE;
                const DeletePolicyType deletePolicy =
                       itut_x780::deleteOnlyIfNoContainedObjects;
                const string  kind = "CesServiceProfile";
        }; // module CesServiceProfile_AtmSubnetwork


/**
```

5.8.15 LatestOccurenceLog

```
*/
        module LatestOccurenceLog_AtmSubnetworkk
        {
                const string  superiorClass =
                       "atmf_m4nw_v2::AtmSubnetwork";
                const booleansuperiorSubclassesAllowed = TRUE;
                const string  subordinateClass =
                       "atmf_m4nw_v2::LatestOccurenceLog";
                const booleansubordinateSubclassesAllowed = TRUE;
                const booleanmanagerCreatesAllowed = FALSE;
```

```
                const DeletePolicyType deletePolicy =
                       itut_x780::notDeletable;
                const string  kind = "latestOccurenceLog";
          }; // module LatestOccurenceLog_AtmSubnetwork

/**
```

5.8.16 GenericTransportTTP

```
*/
          module GenericTransportTTP_ManagedElement
          {
                const string  superiorClass =
                       " itut_m3120::ManagedElement";
                const booleansuperiorSubclassesAllowed = TRUE;
                const string  subordinateClass =
                       "atmf_m4nw_v2::GenericTransportTTP";
                const booleansubordinateSubclassesAllowed = TRUE;
                const booleanmanagerCreatesAllowed = FALSE;
                const DeletePolicyType deletePolicy =
                       itut_x780::notDeletable;
                const string  kind = "GenericTransportTTP";
          }; // module GenericTransportTTP_ManagedElement



/**
```

5.8.17 ThresholdData

```
*/
          module ThresholdData_Subnetwork
          {
                const string  superiorClass =
                       "atmf_m4nw_v2::AtmSubnetwork";
                const booleansuperiorSubclassesAllowed = TRUE;
                const string  subordinateClass =
                       "itut_q822d1::ThresholdData";
                const booleansubordinateSubclassesAllowed = TRUE;
                const booleanmanagerCreatesAllowed = TRUE;
                const DeletePolicyType deletePolicy =
                       itut_x780::deleteOnlyIfNoContainedObjects;
                const string  kind = "ThresholdData";
          }; // module ThresholdData_Subnetwork




/**
```

5.8.18 CellProtocolMonCD

```
*/
          module CellProtocolMonCD_AtmLinkEnd
          {
                const string  superiorClass =
                       "atmf_m4nw_v2::AtmLinkEnd";
                const booleansuperiorSubclassesAllowed = TRUE;
                const string  subordinateClass =
                       "atmf_m4nw_v2::CellProtocolMonCD";
                const booleansubordinateSubclassesAllowed = TRUE;
                const booleanmanagerCreatesAllowed = TRUE;
                const DeletePolicyType deletePolicy =
```

```
                        itut_x780::deleteOnlyIfNoContainedObjects;
                const string  kind = "CellProtocolMonCD";
        }; // module CellProtocolMonCD_AtmLinkEnd



/**
```

5.8.19 CongDiscardCD

```
*/
        module CongDiscardCD_AtmLinkEnd
        {
                const string  superiorClass =
                        "atmf_m4nw_v2::AtmLinkEnd";
                const booleansuperiorSubclassesAllowed = TRUE;
                const string  subordinateClass =
                        "atmf_m4nw_v2::CongDiscardCD";
                const booleansubordinateSubclassesAllowed = TRUE;
                const booleanmanagerCreatesAllowed = TRUE;
                const DeletePolicyType deletePolicy =
                        itut_x780::deleteOnlyIfNoContainedObjects;
                const string  kind = "CongDiscardCD";
        }; // module CongDiscardCD_AtmLinkEnd


        module CongDiscardCD_AtmSubnetwork
        {
                const string  superiorClass =
                        "atmf_m4nw_v2::AtmSubnetwork";
                const booleansuperiorSubclassesAllowed = TRUE;
                const string  subordinateClass =
                        "atmf_m4nw_v2::CongDiscardCD";
                const booleansubordinateSubclassesAllowed = TRUE;
                const booleanmanagerCreatesAllowed = TRUE;
                const DeletePolicyType deletePolicy =
                        itut_x780::deleteOnlyIfNoContainedObjects;
                const string  kind = "CongDiscardCD";
        }; // module CongDiscardCD_AtmSubnetwork



/**
```

5.8.20 GenericTransportPmCD

```
*/

        module GenericTransportPmCD_AtmLinkEndPhy
        {
                const string  superiorClass =
                        "atmf_m4nw_v2::AtmLinkEndPhy";
                const booleansuperiorSubclassesAllowed = TRUE;
                const string  subordinateClass =
                        "atmf_m4nw_v2:: GenericTransportPmCD";
                const booleansubordinateSubclassesAllowed = TRUE;
                const booleanmanagerCreatesAllowed = TRUE;
                const DeletePolicyType deletePolicy =
                        itut_x780::deleteOnlyIfNoContainedObjects;
                const string  kind = " GenericTransportPmCD";
        }; // module GenericTransportPmCD_AtmLinkEndPhy
```

```
            module GenericTransportPmCD_GenericTransportTTP
            {
                  const string  superiorClass =
                        "atmf_m4nw_v2:: GenericTransportTTP";
                  const booleansuperiorSubclassesAllowed = TRUE;
                  const string  subordinateClass =
                        "atmf_m4nw_v2:: GenericTransportPmCD";
                  const booleansubordinateSubclassesAllowed = TRUE;
                  const booleanmanagerCreatesAllowed = TRUE;
                  const DeletePolicyType deletePolicy =
                        itut_x780::deleteOnlyIfNoContainedObjects;
                  const string  kind = " GenericTransportPmCD";
            }; // module GenericTransportPmCD_GenericTransportTTP



/**
```

5.8.21 PmOamCD

```
*/
            module PmOamCD_AtmNetworkCTP
            {
                  const string  superiorClass =
                        "atmf_m4nw_v2::AtmNetworkCTP";
                  const booleansuperiorSubclassesAllowed = TRUE;
                  const string  subordinateClass =
                        "atmf_m4nw_v2::PmOamCD";
                  const booleansubordinateSubclassesAllowed = TRUE;
                  const booleanmanagerCreatesAllowed = TRUE;
                  const DeletePolicyType deletePolicy =
                        itut_x780::deleteOnlyIfNoContainedObjects;
                  const string  kind = "PmOamCD";
            }; // module PmOamCD_AtmNetworkCTP


            module PmOamCD_AtmNetworkTTP
            {
                  const string  superiorClass =
                        "atmf_m4nw_v2::AtmNetworkTTP";
                  const booleansuperiorSubclassesAllowed = TRUE;
                  const string  subordinateClass =
                        "atmf_m4nw_v2::PmOamCD";
                  const booleansubordinateSubclassesAllowed = TRUE;
                  const booleanmanagerCreatesAllowed = TRUE;
                  const DeletePolicyType deletePolicy =
                        itut_x780::deleteOnlyIfNoContainedObjects;
                  const string  kind = "PmOamCD";
            }; // module PmOamCD_AtmNetworkTTP



/**
```

5.8.22 TcAdaptProtMonCD

```
*/
            module TcAdaptProtMonCD_AtmLinkEnd
            {
                  const string  superiorClass =
                        "atmf_m4nw_v2::AtmLinkEnd";
                  const booleansuperiorSubclassesAllowed = TRUE;
```

```
                const string  subordinateClass =
                       "atmf_m4nw_v2::TcAdaptProtMonCD";
                const booleansubordinateSubclassesAllowed = TRUE;
                const booleanmanagerCreatesAllowed = TRUE;
                const DeletePolicyType deletePolicy =
                       itut_x780::deleteOnlyIfNoContainedObjects;
                const string  kind = "TcAdaptProtMonCD";
        }; // module TcAdaptProtMonCD_AtmLinkEnd


/**
```

5.8.23 TrafficLoadCD

```
*/

        module TrafficLoadCD_AtmLinkEnd
        {
                const string  superiorClass =
                       "atmf_m4nw_v2::AtmLinkEnd";
                const booleansuperiorSubclassesAllowed = TRUE;
                const string  subordinateClass =
                       "atmf_m4nw_v2::TrafficLoadCD";
                const booleansubordinateSubclassesAllowed = TRUE;
                const booleanmanagerCreatesAllowed = TRUE;
                const DeletePolicyType deletePolicy =
                       itut_x780::deleteOnlyIfNoContainedObjects;
                const string  kind = "TrafficLoadCD";
        }; // module TrafficLoadCD_AtmLinkEnd


        module TrafficLoadCD_AtmSubnetwork
        {
                const string  superiorClass =
                       "atmf_m4nw_v2::AtmSubnetwork";
                const booleansuperiorSubclassesAllowed = TRUE;
                const string  subordinateClass =
                       "atmf_m4nw_v2::TrafficLoadCurrentData";
                const booleansubordinateSubclassesAllowed = TRUE;
                const booleanmanagerCreatesAllowed = TRUE;
                const DeletePolicyType deletePolicy =
                       itut_x780::deleteOnlyIfNoContainedObjects;
                const string  kind = "TrafficLoadCD";
        }; // module TrafficLoadCD_AtmSubnetwork


        module TrafficLoadCD_AtmNetworkCTP
        {
                const string  superiorClass =
                       "atmf_m4nw_v2::AtmNetworkCTP";
                const booleansuperiorSubclassesAllowed = TRUE;
                const string  subordinateClass =
                       "atmf_m4nw_v2::TrafficLoadCD";
                const booleansubordinateSubclassesAllowed = TRUE;
                const booleanmanagerCreatesAllowed = TRUE;
                const DeletePolicyType deletePolicy =
                       itut_x780::deleteOnlyIfNoContainedObjects;
                const string  kind = "TrafficLoadCD";
        }; // module TrafficLoadCD_AtmNetworkCTP
```

```
/**
```

5.8.24 UpcNpcDisagreementsCD

```
*/
             module UpcNpcDisagreementsCD_AtmNetworkCTP
             {
                    const string  superiorClass =
                           "atmf_m4nw_v2::AtmNetworkCTP";
                    const booleansuperiorSubclassesAllowed = TRUE;
                    const string  subordinateClass =
                           "atmf_m4nw_v2::UpcNpcDisagreementsCD";
                    const booleansubordinateSubclassesAllowed = TRUE;
                    const booleanmanagerCreatesAllowed = TRUE;
                    const DeletePolicyType deletePolicy =
                           itut_x780::deleteOnlyIfNoContainedObjects;
                    const string  kind = "UpcNpcDisagreementsCD";
             }; // module UpcNpcDisagreementsCD_AtmNetworkCTP


      }; // module NameBinding


}; // module atmf_m4nw_v2


#endif // _atmf_m4nw_v2_idl_
```

```
/**
```

6. Information Model IDL: Constants

```
*/
#ifndef _atmf_m4nwconst_v2_idl_
#define _atmf_m4nwconst_v2_idl_


#pragma prefix "atmforum.com"


/**
This IDL code is intended to be stored in a file named "atmf_m4nwconst_v2.idl"
located in the search path used by IDL compilers on your system.
*/




/**
This module, atmf_m4nw_v2, provides a set of IDL interfaces for managing an
ATM network using the ATM Forum M4 Network View requirements and logical MIB
found in AF-NM-0058.001.  The IDL definitions in this file are only constant
values used within this module. The object interfaces are contained in a
seperate file, but included in the same "atmf_m4nw_v2" moduel.
*/
module atmf_m4nw_v2
{
       const string moduleName = "atmf_m4nw_v2";

/**
```

6.1. *CharacteristicInfoConst*

```
This module contains the constant values defined for the CharacteristicInfo UID.
*/
       module ATMFCharacteristicInfoConst
       {
              const string moduleName =
                     "atmf_m4nw_v2::ATMFCharacteristicInfoConst";

/**
imaGroupCI: object instances that represent an IMA Group
*/
              const short imaGroupCI = 1;



       }; // module ATMFCharacteristicInfoConst



/**
```

6.2. *ATMFProbableCauseConst*

```
This module contains the constant values defined for the ATM specific
ProbableCause UID.
*/
       module ATMFProbableCauseConst
```

```
	{
		const string moduleName =
			"atmf_m4nw_v2::ATMFProbableCauseConst";

		// Loss of Cell Delineation
		const short lCD = 1;

		// PLCP Loss of Frame for DS3
		const short pLCPLOF = 2;

		// PLCP Far End Alarm for DS3
		const short pLCPFE = 3;

		// Communication alarm for ATMSNCReroute
		const short ATMSNCReroute = 4;


		// IMA Link Alarms
		const short ImaAlarmLinkLif = 5;
		const short ImaAlarmLinkLods = 6;
		const short ImaAlarmLinkRfi = 7;
		const short ImaAlarmLinkTxMisConnect = 8;
		const short ImaAlarmLinkRxMisConnect = 9;
		const short ImaAlarmLinkTxFault = 10;
		const short ImaAlarmLinkRxFault = 11;
		const short ImaAlarmLinkTxUnusableFe = 12;
		const short ImaAlarmLinkRxUnusableFe = 13;

		// IMA Group Alarms
		const short ImaAlarmGroupStartupFe = 14;
		const short ImaAlarmGroupCfgAbort = 15;
		const short ImaAlarmGroupCfgAbortFe = 16;
		const short ImaAlarmGroupInsuffLinks = 17;
		const short ImaAlarmGroupInsuffLinksFe = 18;
		const short ImaAlarmGroupBlockedFe = 19;
		const short imaAlarmGroupTimingSynch = 20;

		// AAL Alarms
		const short CellLossIntegrationPeriodExpired = 21;


	}; // module ATMFProbableCauseConst


}; // module atmf_m4nw_v2



#endif // _atmf_m4nw_v2_idl_
```

References

[1] ATM Forum, af-nm-0058.001 ***M4 Interface Requirements and Logical MIB: ATM Network View version 2***, May 1999.

[2] ATM Forum, af-nm-0020.001, ***M4 Interface Requirements and Logical MIB: ATM Network Element View ,*** October 1998.

[3] ITU-T Recommendation X.780, ***TMN Guidelines for Defining CORBA Managed Objects,*** January 2001

[4] ITU-T Recommendation X.780.1, ***TMN Guidelines for Defining Coarse-grained CORBA Managed Objects.,*** August 2001

[5] ITU-T Recommendation Q.816, ***CORBA Based TMN Services,*** January 2001

[6] ITU-T Recommendation Q.816.1, ***CORBA Based TMN Services Extensions to Support Coarse-Grained Interfaces,*** August 2001

[7] ITU-T Recommendation M.3120, ***CORBA Generic Network and NE Level Information Model,*** October 2001

[8] ITU-T Recommendation Q.822.1, ***CORBA-based TMN Performance Management Service,*** October 2001

Appendix A: Object Naming Guidelines

This appendix provides suggested object naming guidelines for the M4 Network View CORBA IDL MIB. Such guidelines will promote EMS and NMS interoperability. The naming guidelines, in the table below, provide a name syntax for each CORBA object. The name syntax should be used as the id of the name component of the object.

Syntax:
"text" – text inside quotation should appear as is
| - means OR
<*type*> - identifies a type or category
[optional item] – indicates an optional item
{repetitive item} – indicates an item that may appear zero or more times

Defined Types used:
<*String*> ::= {any_character}
<*VP_or_VC* > ::= "VP" | "VC" | "VPandVC"
<*NetworkCTPId_or_LinkEndId*> ::= "NetworkCTPId=" | "LinkEndId="
<*Integer-VPI*> ::= <*Integer*>
<*null*> - indicates a null string

Note: asterisk "*" is used as a field delimiter.

Table A-1. Object Naming Guidelines

Object	**Name Syntax** (NameComponent.id for object)
atmf_m4nw_v2:AalProfileTypeOne	"AtmLndLayer="<VP_or_VC>"*" "SubnetworkId="<String> "*" "ProfileId="<String>
atmf_m4nw_v2:AalProfileTypeTwo	"AtmLndLayer="<*VP_or_VC*>"*" "SubnetworkId="<*String*> "*" "ProfileId="<*String*>
atmf_m4nw_v2:AalProfileTypeThreeFour	"AtmLndLayer="<*VP_or_VC*>"*" "SubnetworkId="<*String*> "*" "ProfileId="<*String*>
atmf_m4nw_v2:AalProfileTypeFive	"AtmLndLayer="<*VP_or_VC*>"*" "SubnetworkId="<*String*> "*" "ProfileId="<*String*>
atmf_m4nw_v2:atmImaGroupTP	"LndLayer="IMA"*" "ManagedElementId="<*String*>"*" "Bay="<*String*>"*" "Shelf="<*String*>"*" "Slot="<*String*>"*" "ImaGroupId="<*Integer*>
atmf_m4nw_v2: atmImaLinkTP	"LndLayer="IMA"*" "ManagedElementId="<*String*>"*" "Bay="<*String*>"*" "Shelf="<*String*>"*" "Slot="<*String*>"*"

Object	**Name Syntax** (NameComponent.id for object)
	"Port="<*String*>"*" " ImaLinkId ="<*Integer* >
atmf_m4nw_v2:AtmLink	"AtmLndLayer="<*VP_or_VC*>"*" "ManagedElementId="<*String*>"*" "Bay="<*String*>"*" "Shelf="<*String*>"*" "Slot="<*String*>"*" "Port="<*String*>"*" "AorZLinkEnd="<*String*> **OR** "AtmLndLayer="<*VP_or_VC*>"*" "LinkId=”<*String*>
atmf_m4nw_v2:AtmLinkEnd	"AtmLndLayer="<*VP_or_VC*>"*" "ManagedElementId="<*String*>"*" "Bay="<*String*>"*" "Shelf="<*String*>"*" "Slot="<*String*>"*" "Port="<*String*>"*" "AtmLinkEndId="<*String*>
atmf_m4nw_v2:AtmLinkEndPhy	"AtmLndLayer="<*VP_or_VC*>"*" "ManagedElementId="<*String*>"*" "Bay="<*String*>"*" "Shelf="<*String*>"*" "Slot="<*String*>"*" "Port="<*String*>"*" "AtmLinkEndId="<*String*>
atmf_m4nw_v2:AtmNetworkCTP	"AtmLndLayer="<*VP_or_VC*>"*" "ManagedElementId="<*String*>"*" "Bay="<*String*>"*" "Shelf="<*String*>"*" "Slot="<*String*>"*" "Port="<*String*>"*" "AtmLinkEndId="<*String*>"*" "ATMNetworkCTPId="<*Integer-VPI*>["/"<*Integer-VCI*>]
atmf_m4nw_v2:AtmNetworkTTP	"AtmLndLayer="<*VP_or_VC*>"*" "ManagedElementId="<*String*>"*" "Bay="<*String*>"*" "Shelf="<*String*>"*" "Slot="<*String*>"*" "Port="<*String*>"*" "AtmLinkEndId="<*String*>"*" "ATMNetworkTTPId="<*Integer-VPI*>["/"<*Integer-VCI*>]
atmf_m4nw_v2:AtmSNC	"AtmLndLayer="<*VP_or_VC*>"*" "SubnetworkId="("Root" | <*String*>)"*" "ManagedElementId="<*String*>"*"

Object	**Name Syntax** (NameComponent.id for object)
	"Bay="<*String*>"*" "Shelf="<*String*>"*" "Slot="<*String*>"*" "Port="<*String*>"*" "AorZLinkEndId="<*String*>"*" "AorZNetworkCTPId="<*Integer-VPI*>["/"<*Integer-VCI*>] **OR** "AtmLndLayer="<*VP_or_VC*>"*" "SubnetworkId="<*String*>"*" "SNCId=”<*String*>
atmf_m4nw_v2:AtmSubnetwork	"AtmLndLayer="<*VP_or_VC*>"*" "SubnetworkId=""Root" | <*String*>
atmf_m4nw_v2:AtmNetworkAccessProfile	"AtmLndLayer="<*VP_or_VC*>"*" "SubnetworkId="<*String*>"*" "ProfileId="<*String*>
atmf_m4nw_v2:AtmNetworkAccessProfile Factory	"EmsId="<*String*>"*" "FactoryId="<*String*>
atmf_m4nw_v2:AtmRoutingProfile	"AtmLndLayer="<*VP_or_VC*>"*" "SubnetworkId="<*String*>"*" "ProfileId="<*String*>
atmf_m4nw_v2:AtmRoutingProfile Factory	"EmsId="<*String*>"*" "FactoryId="<*String*>
atmf_m4nw_v2:CesServiceProfile	"AtmLndLayer="<*VP_or_VC*>"*" "SubnetworkId="<*String*>"*" "ProfileId="<*String*>
atmf_m4nw_v2:CesProfile Factory	"EmsId="<*String*>"*" "FactoryId="<*String*>
atmf_m4nw_v2:AtmAbstractTrafficDesc	"AtmLndLayer="<*VP_or_VC*>"*" "SubnetworkId="<*String*>"*" "ProfileId="<*String*>
atmf_m4nw_v2:AtmTrafficDescABR	"AtmLndLayer="<*VP_or_VC*>"*" "SubnetworkId="<*String*>"*" "ProfileId="<*String*>
atmf_m4nw_v2:AtmTrafficDescCBR	"AtmLndLayer="<*VP_or_VC*>"*" "SubnetworkId="<*String*>"*" "ProfileId="<*String*>
atmf_m4nw_v2:AtmTrafficDescVBR	"AtmLndLayer="<*VP_or_VC*>"*" "SubnetworkId="<*String*>"*" "ProfileId="<*String*>
atmf_m4nw_v2:AtmTrafficDescUBR	"AtmLndLayer="<*VP_or_VC*>"*" "SubnetworkId="<*String*>"*" "ProfileId="<*String*>
atmf_m4nw_v2:AtmTrafficDescFactory	"EmsId="<*String*>"*" "FactoryId="<*String*>
atmf_m4nw_v2:GenericTransportTTP	"AtmLndLayer="<*VP_or_VC*>"*" "ManagedElementId="<*String*>"*"

Object	**Name Syntax** (NameComponent.id for object)
	"Bay="<*String*>"*" "Shelf="<*String*>"*" "Slot="<*String*>"*" "Port="<*String*>"*" "GenericTransportTTPId="<*String*>
atmf_m4nw_v2: LatestOccurrenceLog	"AtmLndLayer="<*VP_or_VC*>"*" "SubnetworkId="<*String*> "*" "LatestOccurrenceLogId="<*String*>
atmf_m4nw_v2:CurrentDataFactory	"EmsId="<*String*>"*" "FactoryId="<*String*>
atmf_m4nw_v2:ThresholdData	"AtmLndLayer="<*VP_or_VC*>"*" "SubnetworkId="<*String*>"*" "ThresholdDataId="<*String*>
atmf_m4nw_v2: CellProtocolMonCurrentData	"LinkEndId="<*String*>"*" "CurrentDataId="<*String*>
atmf_m4nw_v2:AtmTrafficLoadCurrent Data	<*NetworkCTPId_or_LinkEndId*><*String*>"*" "CurrentDataId="<*String*>
atmf_m4nw_v2: CongDiscardCurrentData	"LinkEndId="<*String*>"*" "CurrentDataId="<*String*>
atmf_m4nw_v2: TcAdaptProtMonCurrentData	"LinkEndId="<*String*>"*" "CurrentDataId="<*String*>
atmf_m4nw_v2: UpcNpcDisagreementsCurrentData	"NetworkCTPId="<*String*>"*" "CurrentDataId="<*String*>
atmf_m4nw_v2: PmOamCurrentData	"NetworkCTPId="<*String*>"*" "CurrentDataId="<*String*>

Appendix B: UML Diagrams

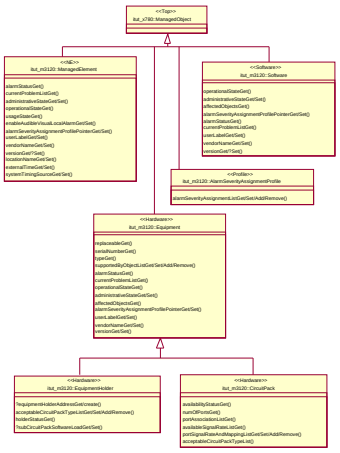


Figure 1 Inheritance Relationship of Equipment Information

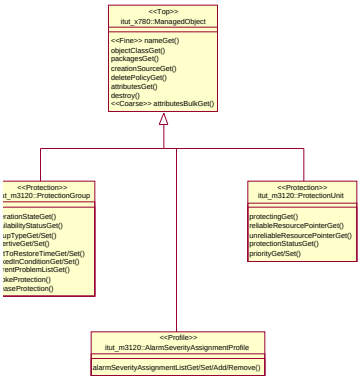


Figure 2 Inheritance Relationship of Protection and Profile Information

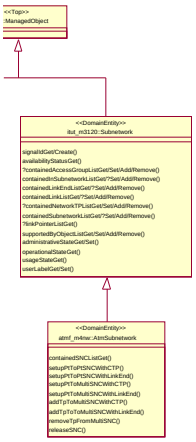


Figure 3 Inheritance Relationship of Domain Entity Information

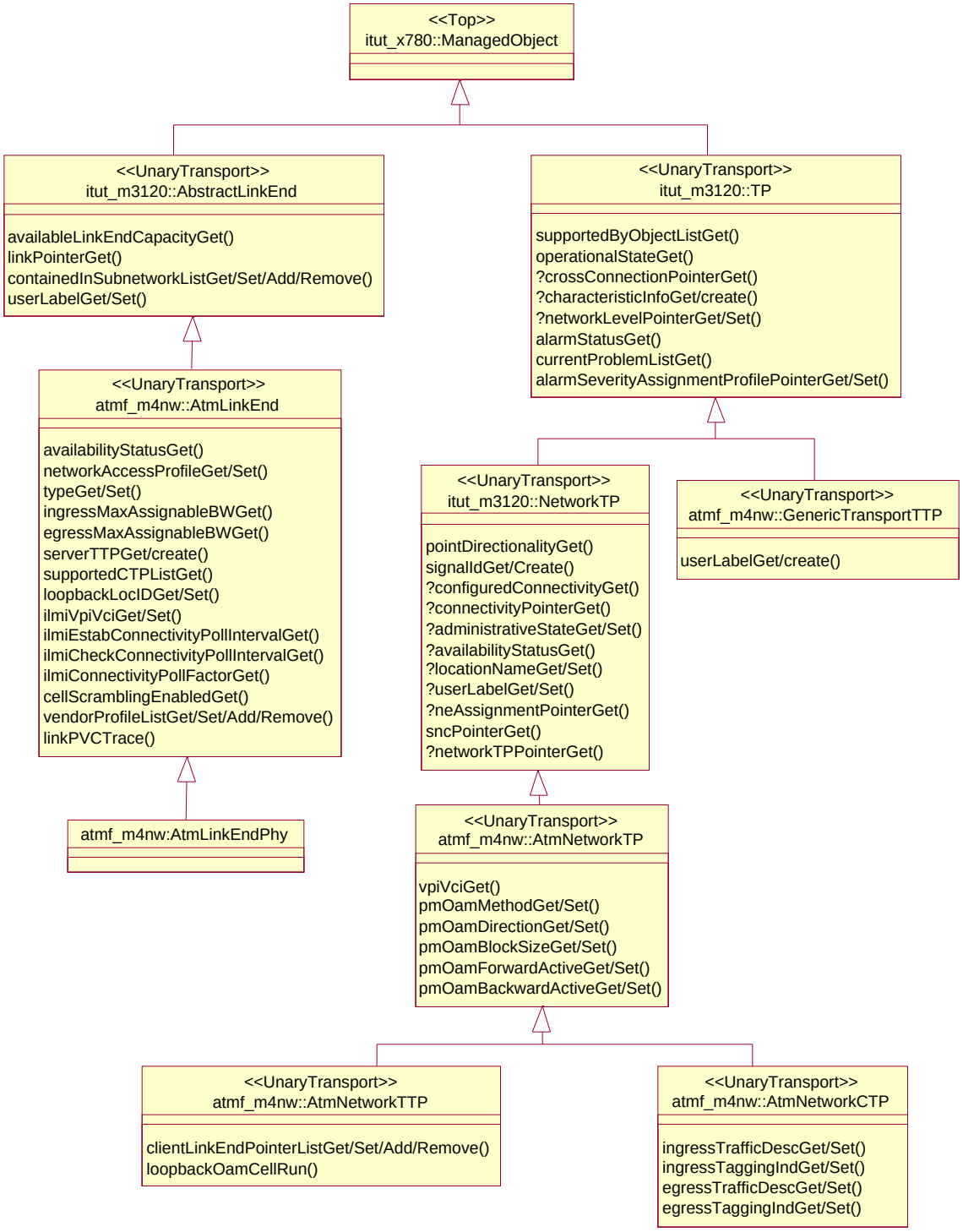


Figure 4 Inheritance Relationship of Unary Transport Information

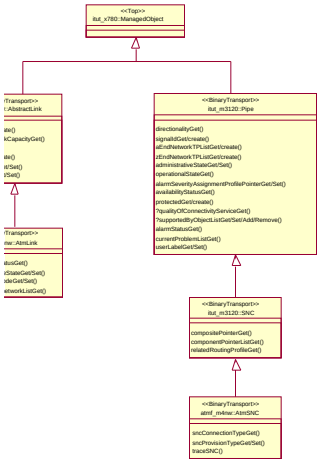


Figure 5 Inheritance Relationship of Binary Transport Information

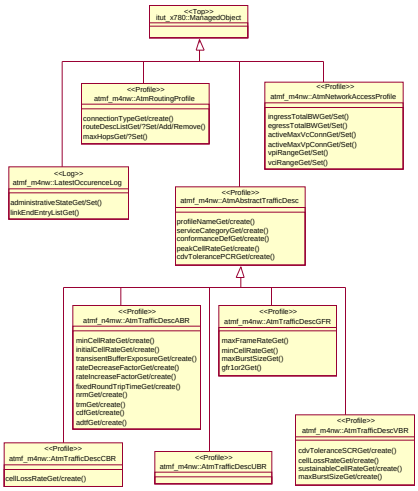


Figure 6 Inheritance Relationship of Profile Information

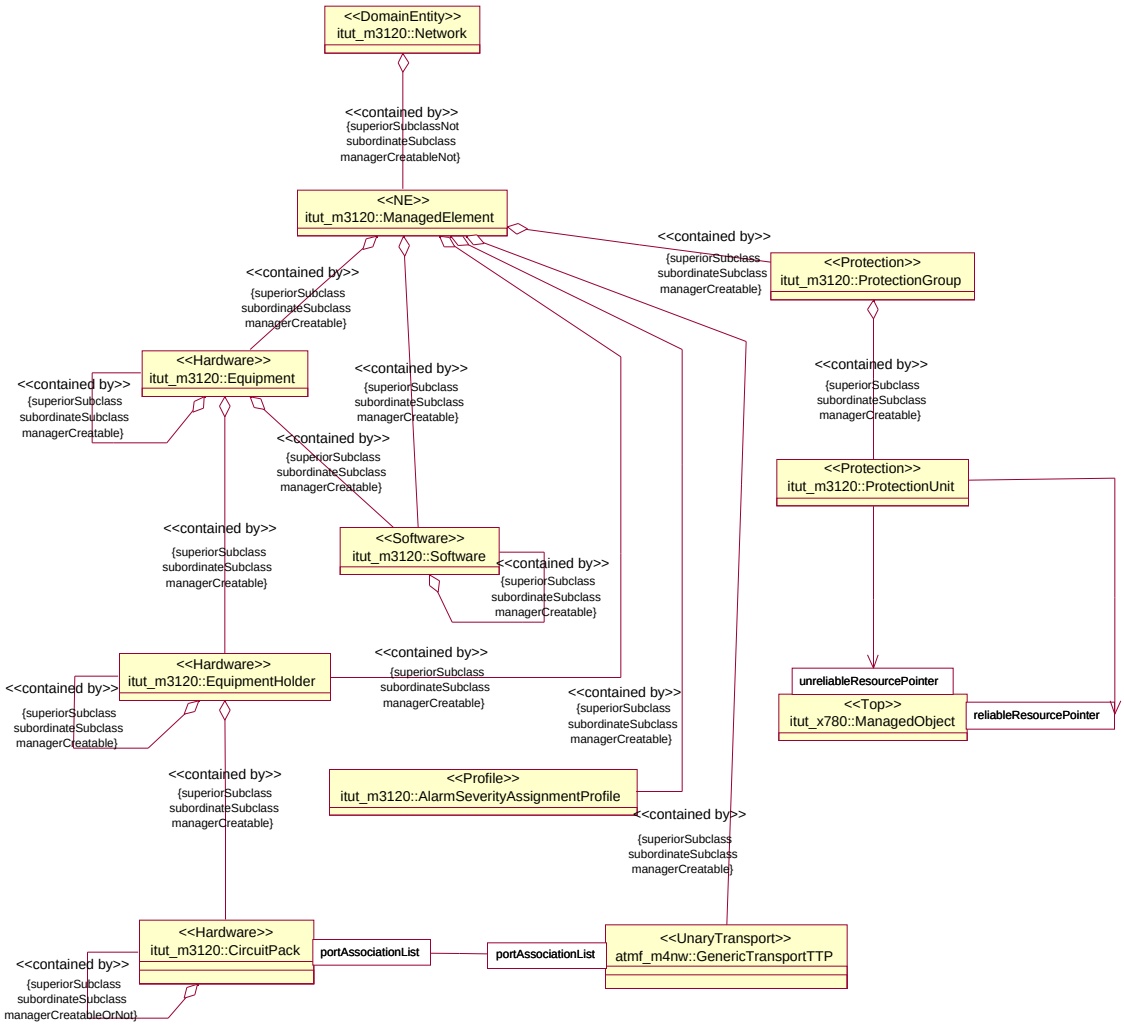


Figure 7 Association Relationship of Equipment Information

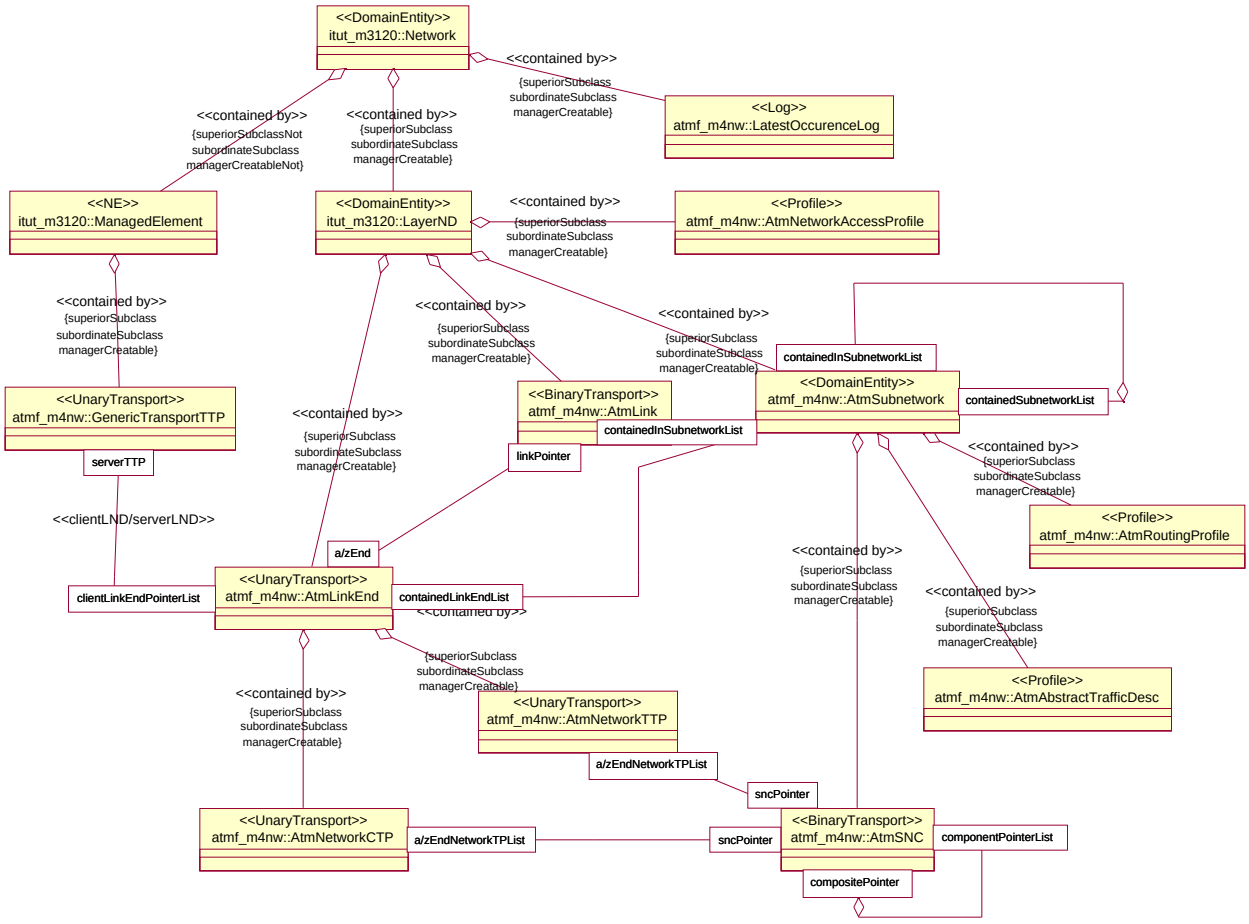


Figure 8 Association Relationship of Connection Information

Appendix C: Instance Diagram Examples

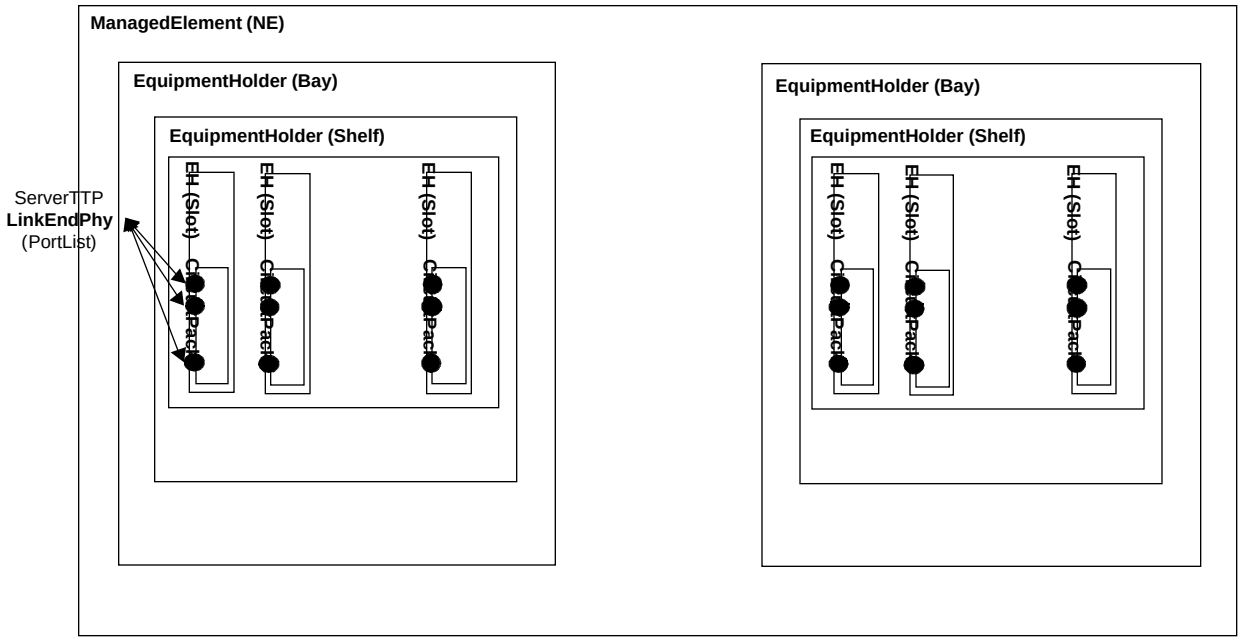


Figure 9 Illustration of Equipment Model

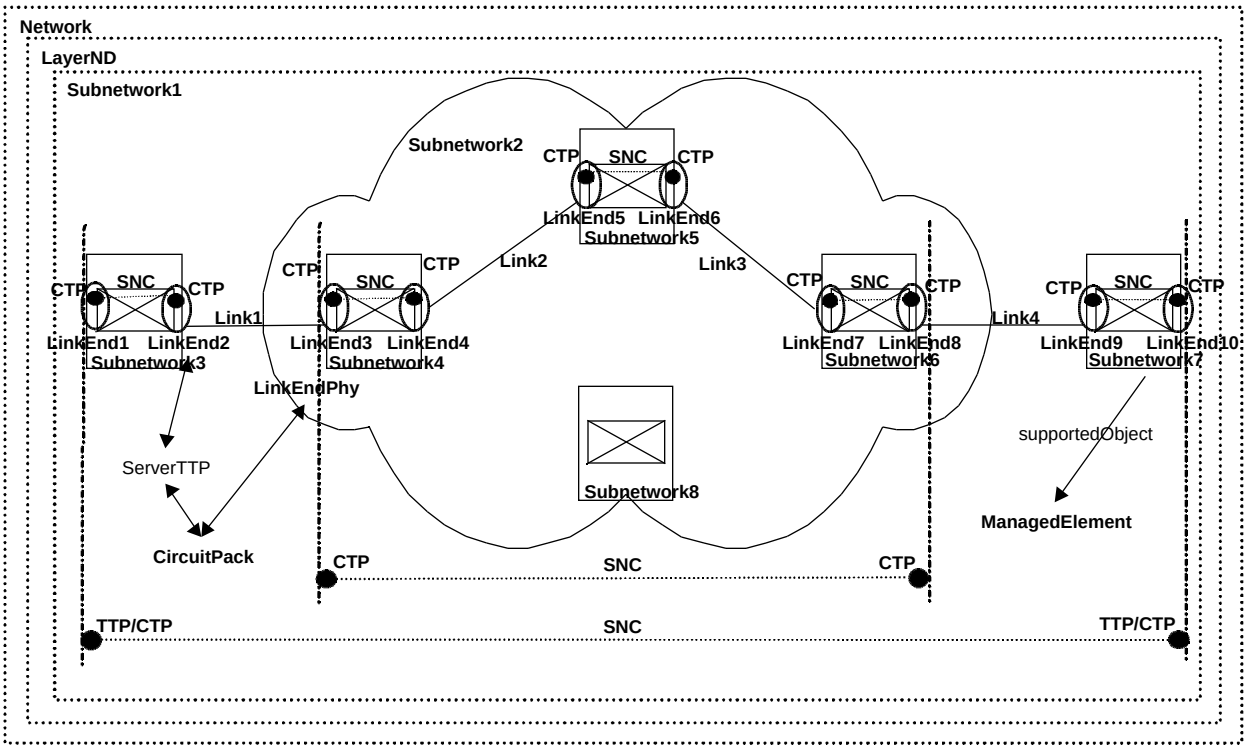


Figure 10 Illustration of Connection Model