ICS 33.040
M 11

YD

中华人民共和国通信行业标准

YD/T 2336.5-2016

分组传送网(PTN)网络管理技术要求 第5部分：基于IDL/IIOP技术的EMS-NMS接口信息模型

Technical requiement for packet transport network (PTN) management Part 5：EMS-NMS interface information modelbased on IDL/IIOP

2016-04-05 发布　　2016-07-01 实施

中华人民共和国工业和信息化部　发布

目　　次

前　言

YD/T 2336-2016《分组传送网（PTN）网络管理技术要求》预计发布如下部分：

- 第 1 部分：基本原则
- 第 2 部分：NMS 系统功能
- 第 3 部分：EMS-NMS 接口功能
- 第 4 部分：EMS-NMS 接口通用信息模型
- 第 5 部分：基于 IDL/IIOP 技术的 EMS-NMS 接口信息模型
- 第 6 部分：基于 XML 技术的 EMS-NMS 接口信息模型

——本部分是 YD/T 2336-2016 的第 5 部分。

本部分按照 GB/T 1.1-2009 给出的规则起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别这些专利的责任。

本部分由中国通信标准化协会提出并归口。

本部分起草单位：中国移动通信集团设计院有限公司、武汉烽火科技集团有限公司、北京邮电大学、华为技术有限公司、北京市天元网络技术股份有限公司、中兴通讯股份有限公司、中国信息通信研究院、上海贝尔股份有限公司、华为技术有限公司、瑞斯康达科技发展股份有限公司。

本部分主要起草人：成梦虹、吕良栋、蒙向阳、刘　娟、吴　翔、芮兰兰、熊　翱、王亚鹏、邓万球、司　昕、徐云斌、张　励、杨　海、张　贺。

分组传送网（PTN）网络管理技术要求 第5部分：基于IDL/IIOP技术的EMS-NMS接口信息模型

1 范围

本部分规定了分组传送网(PTN)网络管理体系EMS-NMS之间基于IDL/IIOP技术的接口规范。

本部分适用于PTN网络管理系统。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅所注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

YD/T 2336.2-2016 分组传送网（PTN）网络管理技术要求 第2部分：NMS系统功能

YD/T 2336.3-2016 分组传送网（PTN）网络管理技术要求 第3部分：EMS-NMS接口功能

YD/T 2336.4-2016 分组传送网（PTN）网络管理技术要求 第4部分：EMS-NMS接口通用信息模型

TMF 814 多技术网络管理解决方案 NML-EML接口 版本3.5（Multi-Technology Network Management Solution Set Document NML-EML Interface Version 3.5）

3 术语、定义和缩略语

3.1 术语和定义

YD/T 2336.2-2016《分组传送网（PTN）网络管理技术要求 第2部分：NMS系统管理功能》和YD/T2336.3-2016《分组传送网（PTN）网络管理技术要求 第3部分：EMS-NMS系统接口功能》中界定的的术语定义适用于本文件。

3.2 缩略语

下列缩略语适用于本文件。

ACL	Access Control List	访问控制列表
AIS	Alarm Indication Signal	告警指示信号
ASAP	Alarm Severity Assignment Profile	告警级别分配模板
ATM	Asynchronous Transfer Mode	异步传输模式
CBR	Constant Bit Rate	承诺突发长度
CC	Continuous Check	连续性监测
CES	Circuit Emulation Service	电路仿真业务
CESoPSN	Structure-aware TDM Circuit Emulation Service over Packet Switched Network	分组网交换承载的结构化电路仿真业务
CIR	Committed Information Rate	承诺信息速率
CoS	Class of Service	业务分类
CTP	Connection Termination Point	连接终端点
CV	Connectivity Verification	连通性验证

DM	Delay Measurement	时延测量
DSCP	DiffServ Code Point	区分业务编码点
EMS	Element Management System	网元管理系统
FDFr	Flow Domain Fragment	流域片段
FTP	Floating Termination Point	浮动终端点
FTP	File Transform Protocol	文件传输协议
ID	Identification	标识符
IMA	Inverse Multiplexing over ATM	ATM反向复用
LAG	Link Aggregation	链路聚合
LB	LoopBack	环回
LT	Link Trace	链路踪迹
MAC	Media Access Control	媒质接入控制
MEG	Maintenance Entity Group	维护实体组
MEP	MEG End Point	MEG端点
MFDFr	Matrix Flow Domain Fragment	矩阵流域片段
MIP	MEG Intermediate Point	MEG中间节点
NMS	Network Management System	网络管理系统
OAM	Operation，Administration and Maintenance	运营、管理和维护
PBS	Peak Burst Size	峰值突发长度
PDH	Plesiochronous Digital Hierarchy	准同步数字体系
PHB	Per-Hop Behavior	每跳行为
PIR	Peak Information Rate	峰值信息速率
PTN	Packet Transport Network	分组传送网
PTP	Physical Termination Point	物理终端点
RDI	Remote Defect Indication	远端缺陷指示
SDH	Synchronous Digital Hierarchy	同步数字体系
SNCP	Sub-Network Connection Protection	子网连接保护
TCA	Threshold Crossing Alert	性能越限告警
TCP/IP	Transmission Control Protocol/Internet Protocol	传输控制协议/互联网协议
TDM	Time Division Multiplex	时分复用
TP	Termination Point	终端点
VC	Virtual Channel	虚通道
VCC	Virtual Channel Connection	虚通道连接
VCI	Virtual Channel Identifier	虚通道标识符
VLAN	Virtual Local Area Network	虚拟局域网
VP	Virtual Path	虚通路
VPC	Virtual Path Connection	虚通路连接

VPI	Virtual Path Identifier	虚通路标识符
VS	Virtual Section	虚段层
WFQ	Weighted Fair Queue	加权公平队列
WRED	Weighted Random Early Detect	加权随机早期检测

4 接口信息描述模板

4.1 数据定义描述模板

定义	
数据结构的 IDL 定义	
说明	
对象说明	数据对象的说明
属性名	属性说明
属性名称	属性含义说明

4.2 接口定义描述模板

定义	
接口操作的 IDL 定义	
说明	
功能描述	接口功能描述说明
输入参数	输入参数含义说明
输入/输出参数	输入/输出参数含义说明
输出参数	输出参数含义说明
操作异常	异常说明

本部分中各模块与IDL文件间的相互关系见附录A。

5 接口规范

5.1 配置管理模块

5.1.1 EMS 管理模块(module emsMgr)

5.1.1.1 EMS（EMS_T）

定义	

struct EMS_T {
globaldefs::NamingAttributes_T name;
string userLabel;
string nativeEMSName;
string owner;
string emsVersion;
string type;
globaldefs::NVSList_T additionalInfo;
};	

说明	
对象说明	EMS 信息
属性名	属性说明
name	EMS 的名称

userLabel	EMS 的用户标签
nativeEMSName	EMS 的本地名称
owner	EMS 的所有者
emsVersion	EMS 的软件版本
type	厂家自定义类型
additionalInfo	附加信息。 包括： — 厂商网管系统类型，包括“EMS”、“SNMS”； — 厂商网管系统 IP 地址及通信端口号； — 厂商网管系统北向接口名称和版本； — 厂商网管系统所在的地理位置（区域、站点、机房，精确到机房）。 — 厂商网管系统运行状态 — 厂商网管系统告警状态 — 厂商网管系统最大网元数目 — 厂商网管系统当前网元数目 — 主备属性（主用，备用） — 所管理设备类型（SDH，WDM，OTN，PTN 设备，或前四个的组合） — 创建者标记：创建人姓名 — 创建日期：工程创建日期 — EMS 所使用的硬件平台信息 — EMS 所使用的软件平台信息 — 通信地址

5.1.1.2 简单过滤条件（SimpleFilterType_T）

定义

struct SimpleFilterType_T {
globaldefs::NamingAttributes_T targetObjName;
ObjectTypeList_T objectTypeList;
ObjectGranularity_T objGranularity;
string ftpAddress;
globaldefs::NVSList_T additionalInfo;
};

说明	
对象说明	EMS 信息
属性名	属性说明
targetObjName	同步的对象范围
objectTypeList	对象类型，例拓扑连接、网元、机架、机框等
objGranularity	同步对象粒度，例如名称、属性或是名称和属性的集合
ftpAddress	FTP 服务器地址
additionalInfo	附加信息。例如文件是否压缩、用户名和密码等信息

5.1.1.3 对象粒度（ObjectGranularity_T）

定义
enum ObjectGranularity_T { CS_NAME,

定义
CS_ATTRIBUTE, CS_FULL }; 说明

对象说明	EMS 信息
属性名	属性说明
CS_NAME	同步的粒度为名称
CS_ATTRIBUTES	同步的粒度为属性
CS_FULL	同步的粒度为名称和属性

5.1.1.4 EMSMgr_I 接口（从 common::Common_I 继承）

5.1.1.4.1 查询 EMS 信息（getEMS）

定义
void getEMS(out EMS_T emsInfo) raises(globaldefs::ProcessingFailureException); 说明

功能描述	NMS 向 EMS 查询 EMS 信息
输入参数	无
输入/输出参数	无
输出参数	emsInfo:：EMS 信息
操作异常	EXCPT_INTERNAL_ERROR

5.1.1.4.2 查询所有顶层子网（getAllTopLevelSubnetworks）

定义
void getAllTopLevelSubnetworks(in unsigned long how_many, out multiLayerSubnetwork::SubnetworkList_T sList, out multiLayerSubnetwork::SubnetworkIterator_I sIt) raises(globaldefs::ProcessingFailureException); 说明

功能描述	查询 EMS 内的所有顶层子网信息。顶层子网指包含物理层且位于子网包含关系的最顶层的子网
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	sList：顶层子网列表。 sIt：迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_TOO_MANY_OPEN_ITERATORS EXCPT_NE_COMM_LOSS

5.1.1.4.3 查询所有顶层子网名称（getAllTopLevelSubnetworkNames）

定义
void getAllTopLevelSubnetworkNames(in unsigned long how_many, out globaldefs::NamingAttributesList_T nameList, out globaldefs::NamingAttributesIterator_I nameIt)

定义	
raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询 EMS 内的所有顶层子网名称
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	nameList：顶层子网名称列表。 nameIt：迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_TOO_MANY_OPEN_ITERATORS EXCPT_NE_COMM_LOSS

5.1.1.4.4 查询所有顶层拓扑连接（getAllTopLevelTopologicalLinks）

定义	
void getAllTopLevelTopologicalLinks(in unsigned long how_many, out topologicalLink::TopologicalLinkList_T topoList, out topologicalLink::TopologicalLinkIterator_I topoIt) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询所有顶层拓扑连接，这里的顶层拓扑连接指 EMS 管理的子网间的拓扑连接
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	topoList：拓扑连接列表。 topoIt：迭代器
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.1.4.5 查询所有顶层拓扑连接名称（getAllTopLevelTopologicalLinkNames）

定义	
void getAllTopLevelTopologicalLinkNames(in unsigned long how_many, out globaldefs::NamingAttributesList_T nameList, out globaldefs::NamingAttributesIterator_I nameIt) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询所有顶层拓扑连接的名称
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	nameList：拓扑连接名称列表。 nameIt：迭代器
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.1.4.6 查询指定的顶层拓扑连接（getTopLevelTopologicalLink）

定义	
void getTopLevelTopologicalLink(in globaldefs::NamingAttributes_T topoLinkName, out topologicalLink::TopologicalLink_T topoLink) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	指定拓扑连接名称查询拓扑连接信息
输入参数	topoLinkName：拓扑链接名称
输入/输出参数	无
输出参数	topoLink：拓扑连接
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND

5.1.1.4.7 查询所有当前告警（getAllEMSAndMEActiveAlarms）

定义	
void getAllEMSAndMEActiveAlarms(in notifications::ProbableCauseList_T excludeProbCauseList, in notifications::PerceivedSeverityList_T excludeSeverityList, in unsigned long how_many, out notifications::EventList_T eventList, out notifications::EventIterator_I eventIt) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询所有当前告警（包括越门限告警），包括所有网元以及 EMS 本身产生的告警。查询条件包括告警原因和告警级别
输入参数	excludeProbCauseList：表示被排除的告警级别列表。如果列表是空，表示不排除。 excludeSeverityList：表示被排除的告警原因列表。如果列表是空，表示不排除。 how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	eventList：符合条件的当前告警列表。 eventIt：迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.1.4.8 查询 EMS 所有当前告警（getAllEMSSystemActiveAlarms）

定义
void getAllEMSSystemActiveAlarms(in notifications::PerceivedSeverityList_T excludeSeverityList, in unsigned long how_many, out notifications::EventList_T eventList, out notifications::EventIterator_I eventIt) raises(globaldefs::ProcessingFailureException);

定义	
说明	
功能描述	查询所有 EMS 本身产生的当前告警
输入参数	excludeSeverityList：表示被排除的告警原因列表，如果列表是空，表示不排除。 how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	eventList：符合条件的当前告警列表。 eventIt：迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.1.4.9 创建告警级别分配模板（createASAP）(可选)

定义	
void createASAP(in aSAP::ASAPCreateModifyData_T newASAPCreateData, out aSAP::ASAP_T newASAP, out globaldefs::NVSList_T additionalInfo) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	创建告警级别分配模板
输入参数	newASAPCreateData：ASAP 创建数据
输入/输出参数	无
输出参数	newASAP：新创建的 ASAP。 additionalInfo：附加信息
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INVALID_INPUT EXCPT_INTERNAL_ERROR EXCPT_UNABLE_TO_COMPLY

5.1.1.4.10 删除告警级别分配模板（deleteASAP）(可选)

定义	
void deleteASAP(in globaldefs::NamingAttributes_T aSAPName, inout globaldefs::NVSList_T additionalInfo) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	删除告警级别分配模板
输入参数	aSAPName：指定的要删除的告警级别分配模板名称
输入/输出参数	additionalInfo：附加信息
输出参数	无
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_OBJECT_IN_USE EXCPT_UNABLE_TO_COMPLY

5.1.1.4.11 分配告警级别分配模板（assignASAP）(可选)

定义	
void assignASAP(in globaldefs::NamingAttributes_T aSAPName, in globaldefs::NamingAttributes_T resourceName, in transmissionParameters::LayerRate_T layerRate, inout globaldefs::NVSList_T additionalInfo) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	分配告警级别分配模板
输入参数	aSAPName：告警级别分配模板名称。 resourceName：将要分配 ASAP 模板的资源名称。 layerRate：分配 ASAP 模板的资源的速率
输入/输出参数	additionalInfo：附加信息
输出参数	无
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY

5.1.1.4.12 去分配告警级别分配模板（deassignASAP）(可选)

定义	
void deassignASAP(in globaldefs::NamingAttributes_T resourceName, in transmissionParameters:: LayerRate_T layerRate, inout globaldefs::NVSList_T additionalInfo) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	去分配告警级别分配模板
输入参数	resourceName：将要去分配 ASAP 模板的资源名称。 layerRate：去分配 ASAP 模板的资源的速率
输入/输出参数	additionalInfo：附加信息
输出参数	无
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY

5.1.1.4.13 修改告警级别分配模板（modifyASAP）(可选)

定义
void modifyASAP(in globaldefs::NamingAttributes_T aSAPName, in aSAP::ASAPCreateModifyData_T aSAPModifyData,

定义	
out aSAP::ASAP_T newASAP, out globaldefs::NVSList_T additionalInfo) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	修改告警级别分配模板
输入参数	aSAPName：告警级别分配模板名称。 aSAPModifyData：告警级别分配模板修改数据
输入/输出参数	无
输出参数	newASAP：修改后的告警级别分配模板。 additionalInfo：附加信息
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY

5.1.1.4.14 查询所有告警级别分配模板（getAllASAPs）(可选)

定义	
void getAllASAPs(in unsigned long how_many, out aSAP::ASAPList_T aSAPList, out aSAP::ASAPIterator_I asapIt) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询所有告警级别分配模板
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	aSAPList：告警级别分配模板列表。 asapIt：迭代器
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.1.4.15 查询所有告警级别分配模板名称（getAllASAPNames）(可选)

定义	
void getAllASAPNames(in unsigned long how_many, out globaldefs::NamingAttributesList_T nameList, out globaldefs::NamingAttributesIterator_I nameIt) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询所有告警级别分配模板名称
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无

说明	
输出参数	nameList：名称列表。 nameIt：迭代器
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.1.4.16 查询指定的告警级别分配模板（getASAP）(可选)

定义	
void getASAP(in globaldefs::NamingAttributes_T aSAPName, out aSAP::ASAP_T anASAP) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	指定告警级别分配模板名称，查询告警级别分配模板
输入参数	aSAPName：告警级别分配模板名称
输入/输出参数	无
输出参数	anASAP：告警级别分配模板
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND

5.1.1.4.17 查询与指定资源匹配的告警级别分配模板（getASAPbyResource）(可选)

定义	
void getASAPbyResource(in globaldefs::NamingAttributes_T resourceName, in transmissionParameters::LayerRateList_T layerRateList, in unsigned long how_many, out aSAP::ASAPList_T aSAPList, out aSAP::ASAPIterator_I asapIt) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询与指定资源匹配的告警级别分配模板
输入参数	resourceName：资源名称。 layerRateList：层速率列表。 how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	aSAPList：告警级别分配模板列表。 asapIt：迭代器
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.1.4.18 查询告警级别分配模板相关资源的名称（getASAPAssociatedResourceNames）(可选)

定义	
void getASAPAssociatedResourceNames (in globaldefs::NamingAttributes_T objectName, out globaldefs::NamingAttributesList_T resourceNameList) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	指定对象名称，查询该对象的告警级别分配模板相关资源的名称
输入参数	objectName：对象名称
输入/输出参数	无
输出参数	resourceNameList：查询获得的资源名称列表
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.1.1.4.19 同步配置信息（getInventoryEx）

定义	
void getInventoryEx (in SimpleFilterType_T filter) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	NMS 对厂商网管系统的全部配置数据进行同步，并可指定同步对象类型
输入参数	filter：同步配置信息过滤条件
输入/输出参数	无
输出参数	无
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT

5.1.1.4.20 配置信息同步 ByPull

定义	
无	
说明	
功能描述	EMS 首先定期生成全网配置数据，并放在 FTP 服务器上。然后通过拉模式的文件传送通知知会 NMS 文件已准备好。最后 NMS 根据文件传送通知中指定的 FTP 位置获取到全网配置数据，进行全网配置数据的同步
输入参数	无
输入/输出参数	无
输出参数	全网配置数据文件命名见附录 B。 周期性生成全网配置数据文件； 配置数据文件准备周期为 24h，保留 3 个周期的配置数据文件
操作异常	无

5.1.1.4.21 获取网管历史告警(getHistoryAlarmDataByPull)

定义
void getHistoryAlarmDataByPull(in globaldefs::NamingAttributesList_T nameList, in string taskName, in string compressType, in string packingType, in notifications::ProbableCauseList_T excludeProbCauseList, in notifications::PerceivedSeverityList_T excludeSeverityList, in globaldefs::Time_T startTime, in globaldefs::Time_T endTime) raises(globaldefs::ProcessingFailureException);

说明	
功能描述	获取网管历史告警
输入参数	nameList：指定同步的范围，如 EMS、网元等。 taskName：NMS 指定的任务名称，由 NMS 自定义并保证唯一性 compressType：指定文件是否压缩以及压缩包的类型（取值为：NO_COMPRESSION， GZIP）” packingType：指定文件是否打包以及打包的方式（取值为：NO_PACKING，ZIP，TAR）” excludeProbCauseList：被排除的告警原因列表，如果列表为空，表示查询所有原因的告警 excludeSeverityList：被排除的告警严重性列表，如果列表为空，表示查询所有严重性等级的告警 startTime：告警的起始时间，告警发生或告警消除的网元时间(NeTime)大于等于发生时间的告警才会同步 endTime：告警的终止时间，告警发生或告警消除的网元时间(NeTime)小于等于终止时间的告警才会被同步
输入/输出参数	无
输出参数	全网历史告警文件命名见附录 B。 历史告警文件的保留时间为 3 天（72h）
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.1.2 拓扑连接模块（module topologicalLink）

5.1.2.1 拓扑连接（TopologicalLink_T）

定义
struct TopologicalLink_T { globaldefs::NamingAttributes_T name; string userLabel; string nativeEMSName; string owner; globaldefs::ConnectionDirection_T direction; transmissionParameters::LayerRate_T rate;

定义
globaldefs::NamingAttributes_T aEndTP; globaldefs::NamingAttributes_T zEndTP; globaldefs::NVSList_T additionalInfo; };
说明

对象说明	拓扑连接信息
属性名	属性说明
name	拓扑连接名称
userLabel	拓扑连接的用户标签
nativeEMSName	拓扑连接的本地名称
owner	拓扑连接的所有者
direction	拓扑连接的方向
rate	拓扑连接的速率
aEndTP	拓扑连接的 A 点名称
zEndTP	拓扑连接的 Z 点名称
additionalInfo	附加信息。 包括： — 子类型，包括“网元内拓扑连接”、“网元间拓扑连接”、“跨 EMS 拓扑连接” (可选)

5.1.2.2 拓扑连接列表（TopologicalLinkList_T）

定义
typedef sequence<TopologicalLink_T> TopologicalLinkList_T;
说明

对象说明	拓扑连接列表

5.1.2.3 TopologicalLinkIterator_I 接口

5.1.2.3.1 从迭代器中查询数据（next_n）

定义
boolean next_n(in unsigned long how_many, out TopologicalLinkList_T topoLinkList) raises (globaldefs::ProcessingFailureException);
说明

功能描述	从迭代器中查询 *n* 条记录。当 how_many≥迭代器剩余记录数时，系统在数据取完后应该自动销毁该迭代器对象
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	topoLinkList：首次查询返回的拓扑连接列表
操作异常	EXCPT_INTERNAL_ERROR

5.1.2.3.2 查询迭代器记录条数（getLength）

定义
unsigned long getLength() raises (globaldefs::ProcessingFailureException);

说明	
功能描述	查询迭代器中数据记录的总条数（注意这里指的是迭代器中数据记录的总条数，该值在迭代器生命周期内是不变的）
输入参数	无
输入/输出参数	无
输出参数	无
返回值	unsigned long：迭代器中包含的总的记录条数
操作异常	EXCPT_INTERNAL_ERROR

5.1.2.3.3 销毁迭代器对象（destroy）

定义
void destroy() raises (globaldefs::ProcessingFailureException);

说明	
功能描述	销毁迭代器对象
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR

5.1.3 网元模块（module managedElement）

5.1.3.1 通信状态（CommunicationState_T）

定义

enum CommunicationState_T {
CS_AVAILABLE,
CS_UNAVAILABLE
};

说明	
对象说明	表示系统或模块之间（如 EMS 与网元之间）的通信状态
类型取值	取值说明
CS_AVAILABLE	通讯正常
CS_UNAVAILABLE	通讯失效

5.1.3.2 网元（ManagedElement_T）

定义
struct ManagedElement_T { globaldefs::NamingAttributes_T name; string userLabel; string nativeEMSName; string owner; string location; string version; string productName; CommunicationState_T communicationState; boolean emsInSyncState; transmissionParameters::LayerRateList_T supportedRates;

定义	
globaldefs::NVSList_T additionalInfo; };	
说明	
对象说明	网元信息
属性名	属性说明
name	网元名称
userLabel	网元的用户标签
nativeEMSName	网元的本地名称
owner	网元的所有者
location	网元的地理位置
version	网元的软件版本
productName	网元型号
communicationState	网元的运行状态
emsInSyncState	表示 EMS 是否能够保持 EMS 数据与网元数据同步，并且能够发送所有适当的通知。当 EMS 将此属性设置为假时，表示 EMS 需要与网元进行数据同步并且不能产生适当的通知（如对象创建、删除、属性改变通知）；当 EMS 将此属性设置回到真表示数据重新同步结束同时通知可以在适当的时候产生
supportedRates	网元支持的交叉连接速率
additionalInfo	附加信息。 包括： — 子类型，包括：OADM、OTM、OLA、虚拟网元等(可选)。 — 网元的硬件版本(可选)。 — 网元供应商名称。 — 网元 IP 地址。 — 创建者标记。 — 创建日期

5.1.3.3 网元列表（ManagedElementList_T）

定义	
typedef sequence<ManagedElement_T> ManagedElementList_T;	
说明	
对象说明	网元列表

5.1.3.4 设备时钟(ClockSource_T)

定义
struct ClockSource_T { globaldefs::NamingAttributes_T name; string userLabel; string nativeEMSName; string owner; string clockStatus; string synchroSrcClass; string synchroSrc; string workMode;

定义
stringList_T clockSourceList; sring synchroProtocol; globaldefs::NVSList_T additionalInfo; };

说明	
对象说明	网元信息
属性名	属性说明
name	设备时钟名称
userLabel	设备时钟的用户标签
nativeEMSName	设备时钟的本地名称
owner	设备时钟的所有者
clockStatus	设备时钟状态
synchroSrcClass	设备时钟定时方式，如“内部时钟源”、“线路时钟源”等
synchroSrc	设备厂商跟踪的具体的时钟源
workMode	时钟工作模式
clockSourceList	时钟源优先级列表
synchroProtocol	主要参考源类型，如“1588V2”、“同步以太”等
additionalInfo	附加信息

5.1.3.5 设备时钟列表（ClockSourceList_T）

定义	
typedef sequence<ClockSource_T> ClockSourceList_T;	
说明	
对象说明	设备时钟列表

5.1.3.6 ManagedElementIterator_I 接口

5.1.3.6.1 从迭代器中查询数据（next_n）

定义	
boolean next_n(in unsigned long how_many, out ManagedElementList_T meList) raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	从迭代器中查询 *n* 条记录。当 how_many≥迭代器剩余记录数时，系统在数据取完后应该自动销毁该迭代器对象
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	meList：首次查询返回的网元列表
操作异常	EXCPT_INTERNAL_ERROR

5.1.3.6.2 查询迭代器记录条数（getLength）

定义
unsigned long getLength() raises (globaldefs::ProcessingFailureException);

说明	
功能描述	查询迭代器中数据记录的总条数（注意这里指的是迭代器中数据记录的总条数，该值在迭代器生命周期内是不变的）
输入参数	无
输入/输出参数	无
输出参数	无
返回值	unsigned long：迭代器中包含的总的记录条数
操作异常	EXCPT_INTERNAL_ERROR

5.1.3.6.3 销毁迭代器对象（destroy）

定义	
void destroy() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	销毁迭代器对象
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR

5.1.3.7 设备时钟迭代器 ClockIteratro_I 接口

5.1.3.7.1 从迭代器中查询数据(next_n)

定义	
boolean next_n(in unsigned long how_many, out ClockList_T clockList) raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	从迭代器中查询 *n* 条记录。当 how_many≥迭代器剩余记录数时，系统在数据取完后应该自动销毁该迭代 器对象
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	clockList：首次查询返回的设备时钟列表
操作异常	EXCPT_INTERNAL_ERROR

5.1.3.7.2 查询迭代器记录条数（getLength）

定义	
unsigned long getLength() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	查询迭代器中数据记录的总条数（注意这里指的是迭代器中数据记录的总条数，该值在迭代器生命周期内是不变的）
输入参数	无
输入/输出参数	无
输出参数	无
返回值	unsigned long：迭代器中包含的总的记录条数
操作异常	EXCPT_INTERNAL_ERROR

5.1.3.7.3 销毁迭代器对象（destory）

定义	
void destroy() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	销毁迭代器对象
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR

5.1.3.8 时钟链路迭代器 TopologicalLinkIterator_I 接口

5.1.3.8.1 从迭代器中查询数据（next_n）

定义	
boolean next_n(in unsigned long how_many, out TopologicalLinkList_T clockLinkList) raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	从迭代器中查询 *n* 条记录。当 how_many≥迭代器剩余记录数时，系统在数据取完后应该自动销毁该迭代器对象
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	meList：首次查询返回的时钟链路网络列表
操作异常	EXCPT_INTERNAL_ERROR

5.1.3.8.2 查询迭代器对象（getLength）

定义	
unsigned long getLength() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	查询迭代器中数据记录的总条数（注意这里指的是迭代器中数据记录的总条数，该值在迭代器生命周期内是不变的）
输入参数	无
输入/输出参数	无
输出参数	无
返回值	unsigned long：迭代器中包含的总的记录条数
操作异常	EXCPT_INTERNAL_ERROR

5.1.3.8.3 销毁迭代器对象（destory）

定义
void destroy() raises (globaldefs::ProcessingFailureException);

说明	
功能描述	销毁迭代器对象
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR

5.1.4 网元管理模块（module managedElementManager）

5.1.4.1 ManagedElementMgr_I 接口（从 common::Common_I 继承）

5.1.4.1.1 查询 EMS 下所有网元（getAllManagedElements）

定义	
void getAllManagedElements(in unsigned long how_many, out managedElement::ManagedElementList_T meList, out managedElement::ManagedElementIterator_I meIt) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询 EMS 下所有网元
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	meList：网元列表。 meIt：迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.4.1.2 查询 EMS 下所有网元名称（getAllManagedElementNames）

定义	
void getAllManagedElementNames(in unsigned long how_many, out globaldefs::NamingAttributesList_T nameList, out globaldefs::NamingAttributesIterator_I nameIt) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询 EMS 下所有网元名称
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	nameList：名称列表。 nameIt：迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.4.1.3 查询指定的网元（getManagedElement）

定义
void getManagedElement(in globaldefs::NamingAttributes_T managedElementName, out managedElement::ManagedElement_T me)

定义	
raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	指定名称查询网元
输入参数	managedElementName：网元名称
输入/输出参数	无
输出参数	me：网元信息
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

5.1.4.1.4 查询网元下所有的 PTP 信息（getAllPTPs）

定义	
void getAllPTPs(in globaldefs::NamingAttributes_T managedElementName, in transmissionParameters::LayerRateList_T tpLayerRateList, in transmissionParameters::LayerRateList_T connectionLayerRateList, in unsigned long how_many, out terminationPoint::TerminationPointList_T tpList, out terminationPoint::TerminationPointIterator_I tpIt) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询网元下所有的 PTP 信息，包括物理终端点、浮动终端点，其中浮动终端点包括 L3VPN 中的二层虚接口、三层虚接口
输入参数	managedElementName：网元名称。 tpLayerRateList: 要求满足的 PTP 速率等级列表，查询结果的 PTP 的层速率至少要符合列表中的一项，如果是空，则没有限制。 connectionLayerRateList：要求满足的连接层速率等级列表，查询出的终端点的连接层速率至少要符合列表中的一项，如果是空，则没有限制。 how_many：首次迭代查询的数目
输入/输出参数	无
输出参数	tpList：查询获得的终端点列表。 tpIt：迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.4.1.5 查询网元下所有的 PTP 名称（getAllPTPNames）

定义
void getAllPTPNames(in globaldefs::NamingAttributes_T managedElementName, in transmissionParameters::LayerRateList_T tpLayerRateList,

定义	
in transmissionParameters::LayerRateList_T connectionLayerRateList, in unsigned long how_many, out globaldefs::NamingAttributesList_T nameList, out globaldefs::NamingAttributesIterator_I nameIt) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询网元下所有的 PTP 名称
输入参数	managedElementName：网元名称。 tpLayerRateList：要求满足的 PTP 速率等级列表，查询返回名称的 PTP 的层速率至少要符合列表中的一项，如果是空，则没有限制。 connectionLayerRateList：要求满足的连接层速率等级列表，查询出名称的终端点的连接层速率至少要符合列表中的一项，如果是空，则没有限制。 how_many：首次迭代查询的数目
输入/输出参数	无
输出参数	nameList：名称列表。 nameIt：迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.4.1.6 查询指定的 TP（getTP）

定义	
void getTP(in globaldefs::NamingAttributes_T tpName, out terminationPoint::TerminationPoint_T tp) raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	指定名称查询 TP 信息
输入参数	tpName：TP 名称
输入/输出参数	无
输出参数	tp：TP 信息
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

5.1.4.1.7 查询所属 TP 信息（getContainingTPs）

定义
void getContainingTPs(in globaldefs::NamingAttributes_T tpName, out terminationPoint::TerminationPointList_T tpList) raises (globaldefs::ProcessingFailureException);

说明	
功能描述	查询所属 TP 信息
输入参数	tpName：TP 名称
输入/输出参数	无
输出参数	tpList：TP 列表
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

5.1.4.1.8 查询所属 TP 名称（getContainingTPNames）

定义
void getContainingTPNames(in globaldefs::NamingAttributes_T tpName, out globaldefs::NamingAttributesList_T tpNameList) raises (globaldefs::ProcessingFailureException);

说明	
功能描述	查询所属 TP 名称
输入参数	tpName：TP 名称
输入/输出参数	无
输出参数	tpNameList：TP 名称列表
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

5.1.4.1.9 查询包含的当前的 TP 信息（getContainedCurrentTPs）

定义
void getContainedCurrentTPs(in globaldefs::NamingAttributes_T tpName, in transmissionParameters::LayerRateList_T LayerRateList, in unsigned long::how_many, out terminationPoint::TerminationPointList_T tpList, out terminationPoint::TerminationPointIterator_I tpIt) raises (globaldefs::ProcessingFailureException);

说明	
功能描述	查询包含的当前的 TP 信息，包括连接终端点、L3VPN 下的二层子接口和三层子接口
输入参数	tpName： 1）指定的 TP 名称，可以指定 PTP，FTP 或是 CTP，查询其下所包含的已经做了连接的以及可以做连接的所有的 CTP 信息。 2）指定的网元名称，查询网元下所有三层接口和三层子接口信息。 LayerRateList：指定的速率层次。若为空，表示查询指定 TP 下所包含的当前的 TP 信息；若不为空，表示查询指定的速率层次的当前 TP 信息

说明	
输入/输出参数	无
输出参数	tpList：查询到的当前的 TP 信息列表。 tpIt：迭代器
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

5.1.4.1.10 查询包含的当前的 TP 名称（getContainedCurrentTPNames）

定义
void getContainedCurrentTPNames(in globaldefs::NamingAttributes_T tpName, in transmissionParameters::LayerRateList_T LayerRateList, in unsigned long::how_many, out globaldefs::NamingAttributesList_T nameList, out globaldefs::NamingAttributesIterator_T nameIt) raises (globaldefs::ProcessingFailureException);

说明	
功能描述	查询包含的当前的 TP 名称列表
输入参数	tpName：指定的 TP 名称，可以指定 PTP，FTP 或是 CTP，查询其下所包含的已经做了连接的以及可以做连接的所有的 CTP 名称。 LayerRateList：指定的速率层次。若为空，表示查询指定 TP 下所包含的当前的 TP 名称；若不为空，表示查询指定的速率层次的当前 TP 名称
输入/输出参数	无
输出参数	nameList：查询到的当前的 TP 名称列表。 nameIt：迭代器
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

5.1.4.1.11 设置端口参数（setTPData）

定义
void setTPData (in subnetworkConnection_T::TPData_T tpInfo; out terminationPoint::TerminationPoint_T modifiedTP) raises(globaldefs::ProcessingFailureException););

说明	
功能描述	设置端口的参数信息
输入参数	tpInfo：指定的 TP 信息
输入/输出参数	无

说明	
输出参数	modifiedTP：修改之后的 TP 信息
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

注：部分OAM的操作也可以通过该接口设置。例如：使能/禁止连通性验证、使能/禁止锁定等。

5.1.4.1.12 查询指定网元下所有的当前告警（getAllActiveAlarms）

定义
void getAllActiveAlarms(in globaldefs::NamingAttributes_T meName, in notifications::ProbableCauseList_T excludeProbCauseList, in notifications::PerceivedSeverityList_T excludeSeverityList, in unsigned long how_many, out notifications::EventList_T eventList, out notifications::EventIterator_I eventIt) raises(globaldefs::ProcessingFailureException);

说明	
功能描述	查询指定网元下所有的当前告警
输入参数	meName：网元名称 excludeProbCauseList：表示被排除的告警级别列表，如果列表是空，表示不排除。 excludeSeverityList：表示被排除的告警原因列表，如果列表是空，表示不排除。 how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	eventList：符合条件的当前告警列表 eventIt：迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.4.1.13 查询网络设备时钟源（getAllClockSources）

定义
void getAllClocks(in globaldefs::NamingAttributes_T targetObjectName, in unsigned long how_many, out ClockSourceList_T ecList, out ClockIterator_I ecIt) raises(globaldefs::ProcessingFailureException);

说明	
功能描述	查询网络设备时钟源
输入参数	targetObjectName：查询范围 how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	ecList：指定范围内的网络设备时钟源列表。 ecIt：迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.4.1.14 查询点到点的时钟拓扑链路信息（getClockTopoLink）

定义	
void getClockTopoLink(in globaldefs::NamingAttributes_T targetObjectName, in unsigned long how_many, out TopologicalLink_T clockTopoLink) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询时钟拓扑链路信息
输入参数	targetObjectName：查询范围，例如指定的 ME
输入/输出参数	无
输出参数	clockTopoLink：指定范围内的时钟拓扑链路信息
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.4.1.15 查询网元内所有的交叉连接信息（getAllCrossConnections）

定义	
void getAllCrossConnections(in globaldefs::NamingAttributes_T managedElementName, in transmissionParameters::LayerRateList_T connectionRateList, in unsigned long how_many, out subnetworkConnection::CrossConnectList_T ccList, out subnetworkConnection::CCIterator_I ccIt) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询网元内的交叉连接信息
输入参数	managedElementName：指定的 ME 名称。 connectionRateList：连接速率
输入/输出参数	无

说明	
输出参数	ccList：查询到的交叉连接列表信息。 ccIt：迭代器
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS EXCPT_UNABLE_TO_COMPLY

5.1.4.1.16 查询网元内所有的交叉连接信息（getAllEXCrossConnections）

定义
void getAllEXCrossConnections(in globaldefs::NamingAttributes_T managedElementName, in transmissionParameters::LayerRateList_T connectionRateList, in unsigned long how_many, out EXCrossConnectList_T exccList, out EXCCIterator_I exccIt) raises(globaldefs::ProcessingFailureException);

说明	
功能描述	查询网元内的交叉连接信息
输入参数	managedElementName：指定的 ME 名称。 connectionRateList：连接速率
输入/输出参数	无
输出参数	exccList：查询到的交叉连接列表信息。 exccIt：迭代器
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS EXCPT_UNABLE_TO_COMPLY

5.1.5 设备模块（module equipment）

5.1.5.1 设备类型（EquipmentObjectType_T）

定义	
typedef string EquipmentObjectType_T;	
说明	
对象说明	设备类型，由厂家定义

5.1.5.2 设备类型列表（EquipmentObjectTypeList_T）

定义	
typedef sequence<EquipmentObjectType_T> EquipmentObjectTypeList_T;	
说明	
对象说明	设备类型列表

5.1.5.3 设备运行状态（ServiceState_T）

定义	

enum ServiceState_T {
IN_SERVICE,
OUT_OF_SERVICE,
OUT_OF_SERVICE_BY_MAINTENANCE,
SERV_NA
};	

说明	
对象说明	设备运行状态
类型取值	取值说明
IN_SERVICE	单元盘已经插入到槽位中，并已经根据网管系统的配置方式投入运行
OUT_OF_SERVICE	表示该单元盘已经不能运行，即不能根据网管系统的配置完成指定的功能，而且不能执行相应的管理动作
OUT_OF_SERVICE_BY_MAINTENANCE	该单元盘由于维护的目的而处于非正常工作状态
SERV_NA	单元盘的状态不明确

5.1.5.4 设备容器类型（EquipmentHolderType_T）

定义	
typedef string EquipmentHolderType_T;	
说明	
对象说明	设备容器类型
类型取值	取值说明
"rack"	机架
"shelf"	子架
"sub_shelf"	子子架
"slot"	槽位
"sub_slot"	子槽位

5.1.5.5 设备容器状态（HolderState_T）

定义	

enum HolderState_T {
EMPTY,
INSTALLED_AND_EXPECTED,
EXPECTED_AND_NOT_INSTALLED,
INSTALLED_AND_NOT_EXPECTED,
MISMATCH_OF_INSTALLED_AND_EXPECTED,
UNAVAILABLE,
UNKNOWN
};	

说明	
对象说明	设备容器状态
类型取值	取值说明
EMPTY	空状态（网管未配置设备）

说明	
INSTALLED_AND_EXPECTED	安装了设备且类型符合
EXPECTED_AND_NOT_INSTALLED	期待安装设备但未安装（网管已配置了设备）
INSTALLED_AND_NOT_EXPECTED	安装了设备但未期待（网管未安装）
MISMATCH_OF_INSTALLED_AND_EXPECTED	安装了设备但类型不符合
UNAVAILABLE	不可用
UNKNOWN	无关状态

5.1.5.6 设备（Equipment_T）

定义

```
struct Equipment_T
    {
     globaldefs::NamingAttributes_T      name;
     string                              userLabel;
     string                              nativeEMSName;
     string                              owner;
     boolean                             alarmReportingIndicator;
     ServiceState_T                      serviceState;
     EquipmentObjectType_T               expectedEquipmentObjectType;
     EquipmentObjectType_T               installedEquipmentObjectType;
     string                              installedPartNumber;
     string                              installedVersion;
     string                              installedSerialNumber;
     globaldefs::NVSList_T               additionalInfo;
    };
```

说明	
对象说明	设备（单元盘）信息
类型取值	取值说明
name	设备的名称
userLabel	设备的用户标签
nativeEMSName	设备的 EMS 本地名称
owner	设备的所有者
alarmReportingIndicator	设备的告警上报是否被激活
serviceState	设备的运行状态
expectedEquipmentObjectType	期待的设备类型，如果没有可填为空字符串
installedEquipmentObjectType	已安装的设备类型，如果没有可填为空字符串
installedPartNumber	安装设备的零件编号，如果没有可填为空字符串
installedVersion	安装设备的版本，包括软件版本和硬件版本，如果没有可填为空字符串
installedSerialNumber	安装设备的序列号，如果没有可填为空字符串
additionalInfo	附加信息。 包括： — 子类型，包括接口盘、交叉/交换盘、主控盘、时钟盘、电源盘等； — 单元盘告警状态（当前最高告警级别）

5.1.5.7 设备容器（EquipmentHolder_T）

定义

struct EquipmentHolder_T {
globaldefs::NamingAttributes_T name;
string userLabel;
string nativeEMSName;
string owner;
boolean alarmReportingIndicator;
EquipmentHolderType_T holderType;
globaldefs::NamingAttributes_T expectedOrInstalledEquipment;
EquipmentObjectTypeList_T acceptableEquipmentTypeList;
HolderState_T holderState;
globaldefs::NVSList_T additionalInfo;
};

说明	
对象说明	设备容器信息。设备容器指机架、机框和槽位
类型取值	取值说明
name	设备容器的名称
userLabel	设备容器的用户标签
nativeEMSName	设备容器的 EMS 本地名称
owner	设备容器的所有者
alarmReportingIndicator	设备容器的告警上报是否被激活
holderType	设备容器的类型
expectedOrInstalledEquipment	应安板名称，或者已安板名称。如果是空字符串，表示无应安板
acceptableEquipmentTypeList	可以接受的子设备或者设备容器列表。如果是槽位和子槽位则是必须的，表示槽位可以安装的设备
holderState	设备容器当前的状态
additionalInfo	附加信息。 包括： — 子类型；(可选) — 设备容器版本； — 设备容器所在位置（槽位在子架中的相对位置，序号表示）； — 设备容器供应商名称； — 设备容器序列号

5.1.5.8 设备与容器区分器（EquipmentTypeQualifier_T）

定义

enum EquipmentTypeQualifier_T {
EQT,
EQT_HOLDER
};

说明	
对象说明	设备与容器区分器，在设备与设备容器的联合类型中用于区分设备和设备容器
类型取值	取值说明
EQT	设备
EQT_HOLDER	设备容器

5.1.5.9 设备或设备容器（EquipmentOrHolder_T）

定义	

union EquipmentOrHolder_T switch (EquipmentTypeQualifier_T) {
case EQT: Equipment_T equip;
case EQT_HOLDER: EquipmentHolder_T holder;
};	

说明	
对象说明	设备容器与设备的联合结构
类型取值	取值说明
equip	设备
holder	设备容器

5.1.5.10 设备或设备容器列表（EquipmentOrHolderList_T）

定义	
typedef sequence<EquipmentOrHolder_T> EquipmentOrHolderList_T;	
说明	
对象说明	设备或设备容器列表

5.1.5.11 EquipmentOrHolderIterator_I 接口

5.1.5.11.1 从迭代器中查询数据（next_n）

定义	
boolean next_n(in unsigned long how_many, out EquipmentOrHolderList_T equipmentOrHolderList) raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	从迭代器中查询 *n* 条记录。当 how_many≥迭代器剩余记录数时，系统在数据取完后应该自动销毁该迭代器对象
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	equipmentOrHolderList：首次查询返回的设备或设备容器列表
操作异常	EXCPT_INTERNAL_ERROR

5.1.5.11.2 查询迭代器记录条数（getLength）

定义	
unsigned long getLength() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	查询迭代器中数据记录的总条数（注意这里指的是迭代器中数据记录的总条数，该值在迭代器生命周期内是不变的）
输入参数	无
输入/输出参数	无
输出参数	无
返回值	unsigned long：迭代器中包含的总的记录条数
操作异常	EXCPT_INTERNAL_ERROR

5.1.5.11.3 销毁迭代器对象（destroy）

定义	
void destroy() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	销毁迭代器对象
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR

5.1.5.12 EquipmentInventoryMgr_I 接口（从 common::Common_I 继承）

5.1.5.12.1 开启告警上报开关（setAlarmReportingOn）

定义	
void setAlarmReportingOn(in globaldefs::NamingAttributes_T equipmentOrHolderName) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	开启告警上报开关
输入参数	equipmentOrHolderName：设备或设备容器名称
输入/输出参数	无
输出参数	无
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.1.5.12.2 关闭告警上报开关（setAlarmReportingOff）

定义	
void setAlarmReportingOff(in globaldefs::NamingAttributes_T equipmentOrHolderName) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	关闭告警上报开关
输入参数	equipmentOrHolderName：设备或设备容器名称
输入/输出参数	无
输出参数	无
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.1.5.12.3 查询被直接包含的设备或设备容器（getContainedEquipment）

定义	
void getContainedEquipment (in globaldefs::NamingAttributes_T equipmentHolderName, out EquipmentOrHolderList_T equipmentOrHolderList) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询被直接包含的设备或设备容器
输入参数	equipmentOrHolderName：设备或设备容器名称
输入/输出参数	无
输出参数	equipmentOrHolderList：设备或设备容器列表
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.1.5.12.4 查询指定设备信息（getEquipment）

定义	
void getEquipment(in globaldefs::NamingAttributes_T equipmentOrHolderName, out EquipmentOrHolder_T equip) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询指定设备信息
输入参数	equipmentOrHolderName：设备或设备容器名称
输入/输出参数	无
输出参数	equip：设备或容器信息
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

5.1.5.12.5 查询网元下所有设备或设备容器信息（getAllEquipment）

定义
void getAllEquipment(in globaldefs::NamingAttributes_T meOrHolderName, in unsigned long how_many, out EquipmentOrHolderList_T eqList, out EquipmentOrHolderIterator_I eqIt) raises(globaldefs::ProcessingFailureException);

说明	
功能描述	查询网元下所有设备或设备容器信息
输入参数	meOrHolderName：网元或设备容器名称。 how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	eqList：设备或设备容器列表。 eqIt：迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.5.12.6 查询网元下所有设备或设备容器名称（getAllEquipmentNames）

定义
void getAllEquipmentNames(in globaldefs::NamingAttributes_T meOrHolderName, in unsigned long how_many, out globaldefs::NamingAttributesList_T nameList, out globaldefs::NamingAttributesIterator_I nameIt) raises(globaldefs::ProcessingFailureException);

说明	
功能描述	查询网元下所有设备或设备容器名称
输入参数	meOrHolderName：网元或设备容器名称。 how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	nameList：名称列表。 nameIt：迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.5.12.7 查询 TP 所在的设备（getAllSupportingEquipment）

定义
void getAllSupportingEquipment(in globaldefs::NamingAttributes_T ptpOrMfdName, out EquipmentOrHolderList_T eqList) raises(globaldefs::ProcessingFailureException);

功能描述	查询 TP 所在的设备
输入参数	ptpOrMfdName：TP 或 MFD 名称
输入/输出参数	无
输出参数	eqList：设备列表

说明	
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

5.1.5.12.8 查询 TP 所在的设备名称（getAllSupportingEquipmentNames）

定义	
void getAllSupportingEquipmentNames(in globaldefs::NamingAttributes_T ptpOrMfdName, out globaldefs::NamingAttributesList_T nameList) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询 TP 所在的设备名称
输入参数	ptpOrMfdName：TP 或 MFD 名称
输入/输出参数	无
输出参数	nameList：名称列表
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

5.1.5.12.9 查询设备上支持的 PTP（getAllSupportedPTPs）

定义	
void getAllSupportedPTPs(in globaldefs::NamingAttributes_T equipmentName, in unsigned long how_many, out terminationPoint::TerminationPointList_T tpList, out terminationPoint::TerminationPointIterator_I tpIt) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询设备上支持的 PTP
输入参数	equipmentName：设备名称。 how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	tpList：查询获得的终端点列表。 tpIt：迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.5.12.10 查询设备上支持的 PTP 名称（getAllSupportedPTPNames）

定义
void getAllSupportedPTPNames(in globaldefs::NamingAttributes_T equipmentName, in unsigned long how_many, out globaldefs::NamingAttributesList_T nameList, out globaldefs::NamingAttributesIterator_I nameIt) raises(globaldefs::ProcessingFailureException);
说明
功能描述	查询设备上支持的 PTP 名称
输入参数	equipmentName：设备名称。 how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	nameList：查询获得的终端点名称列表。 nameIt：迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.6 终端点模块(module terminationPoint)

5.1.6.1 方向类型（Directionality_T）

定义

enum Directionality_T {
D_NA,
D_BIDIRECTIONAL,
D_SOURCE,
D_SINK
};

说明
对象说明	终端点方向类型
类型取值	取值说明
D_NA	无关
D_BIDIRECTIONAL	双向
D_SOURCE	源
D_SINK	宿

5.1.6.2 TP 连接状态（TPConnectionState_T）

定义

enum TPConnectionState_T {
TPCS_NA,
TPCS_SOURCE_CONNECTED,
TPCS_SINK_CONNECTED,
TPCS_BI_CONNECTED,
TPCS_NOT_CONNECTED
};

说明	
对象说明	终端点连接状态
类型取值	取值说明
TPCS_NA	无关
TPCS_SOURCE_CONNECTED	终端点的 SOURCE 部分参与了连接
TPCS_SINK_CONNECTED	终端点的 SINK 部分参与了连接
TPCS_BI_CONNECTED	终端点的 SINK 和 SOURCE 都参与了连接
TPCS_NOT_CONNECTED	未连接

5.1.6.3 TP 类型（TPType_T）

定义

enum TPType_T {
TPT_PTP,
TPT_CTP,
TPT_TPPool
};

说明	
对象说明	终端点类型
类型取值	取值说明
TPT_PTP	物理终端点
TPT_CTP	连接终端点
TPT_TPPool	终端点池

5.1.6.4 TP 映射模式（TerminationMode_T）

定义

enum TerminationMode_T {
TM_NA,
TM_NEITHER_TERMINATED_NOR_AVAILABLE_FOR_MAPPING,
TM_TERMINATED_AND_AVAILABLE_FOR_MAPPING
};

说明	
对象说明	TP 映射模式
类型取值	取值说明
TM_NA	无关
TM_NEITHER_TERMINATED_NOR_AVAILABLE_FOR_MAPPING	终端点可以配置交叉连接，但终端点不能进一步展开使用
TM_TERMINATED_AND_AVAILABLE_FOR_MAPPING	终端点的下层终端点可以使用，终端点本身不能配置交叉连接

5.1.6.5 TP 保护相关类型（TPProtectionAssociation_T）

定义

enum TPProtectionAssociation_T {
TPPA_NA,
TPPA_PSR_RELATED
};

说明	
对象说明	TP 保护相关类型
类型取值	取值说明
TPPA_NA	无关
TPPA_PSR_RELATED	与保护相关，即存在与终端点具有保护关系的终端点

5.1.6.6 TP（TerminationPoint_T）

定义

struct TerminationPoint_T {
globaldefs::NamingAttributes_T name;
string userLabel;
string nativeEMSName;
string owner;
globaldefs::NamingAttributes_T ingressTrafficDescriptorName;
globaldefs::NamingAttributes_T egressTrafficDescriptorName;
TPType_T type;
TPConnectionState_T connectionState;
TerminationMode_T tpMappingMode;
Directionality_T direction;
transmissionParameters::LayeredParameterList_T transmissionParams;
TPProtectionAssociation_T tpProtectionAssociation;
boolean edgePoint;
globaldefs::NVSList_T additionalInfo;
};

说明	
对象说明	TP 信息
属性名	属性说明
name	TP 名称
userLabel	TP 的用户标签
nativeEMSName	TP 的本地名称
owner	TP 的所有者
ingressTrafficDescriptorName	入业务描述符名称
egressTrafficDescriptorName	出业务描述符名称
type	TP 的类型
connectionState	TP 的连接状态
tpMappingMode	TP 的映射类型
direction	TP 的方向
transmissionParams	层速率及传送参数列表，PTN 相关的层参数，见本标准第 4 部分附录 A 层参数定义

说明	
tpProtectionAssociation	保护相关类型
edgePoint	是否是子网内部拓扑连接的终结点
additionalInfo	附加信息。 包括： — 端口使用状态（空闲/占用/部分占用）

5.1.6.7 TP 列表(TerminationPointList_T)

定义	
typedef sequence<TerminationPoint_T> TerminationPointList_T;	
说明	
对象说明	TP 列表

5.1.6.8 TerminationPointIterator_I 接口

5.1.6.8.1 从迭代器中查询数据（next_n）

定义	
boolean next_n(in unsigned long how_many, out TerminationPointList_T tpList) raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	从迭代器中查询 *n* 条记录。当 how_many≥迭代器剩余记录数时，系统在数据取完后应该自动销毁该迭代器对象
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	tpList：首次查询返回的终端点列表
操作异常	EXCPT_INTERNAL_ERROR

5.1.6.8.2 查询迭代器记录条数（getLength）

定义	
unsigned long getLength() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	查询迭代器中数据记录的总条数（注意这里指的是迭代器中数据记录的总条数，该值在迭代器生命周期内是不变的）
输入参数	无
输入/输出参数	无
输出参数	无
返回值	unsigned long：迭代器中包含的总的记录条数
操作异常	EXCPT_INTERNAL_ERROR

5.1.6.8.3 销毁迭代器对象（destroy）

定义
void destroy() raises (globaldefs::ProcessingFailureException);

说明	
功能描述	销毁迭代器对象
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR

5.1.7 保护模块（module protection）

5.1.7.1 保护策略的锁定状态（ProtectionSchemeState_T）

定义

enum ProtectionSchemeState_T {
PSS_UNKNOWN,
PSS_AUTOMATIC,
PSS_FORCED_OR_LOCKED_OUT
};

说明	
对象说明	保护策略的锁定状态
类型取值	取值说明
PSS_UNKNOWN	无关
PSS_AUTOMATIC	部分锁定
PSS_FORCED_OR_LOCKED_OUT	整个保护组被锁定

5.1.7.2 保护类型（ProtectionType_T）

定义

enum ProtectionType_T{
PT_MSP_APS,
PT_SNCP
};

说明	
对象说明	保护类型
类型取值	取值说明
PT_MSP_APS	复用段保护
PT_SNCP	SNC 保护

5.1.7.3 倒换原因（SwitchReason_T）

定义

enum SwitchReason_T {
SR_NA,
SR_RESTORED,
SR_SIGNAL_FAIL,
SR_SIGNAL_MISMATCH,
SR_SIGNAL_DEGRADE,
SR_AUTOMATIC_SWITCH,
SR_MANUAL
};

说明	
对象说明	倒换原因
类型取值	取值说明
SR_NA	无关
SR_RESTORED	恢复
SR_SIGNAL_FAIL	信号失效
SR_SIGNAL_MISMATCH	信号适配
SR_SIGNAL_DEGRADE	信号裂化
SR_AUTOMATIC_SWITCH	自动倒换
SR_MANUAL	人工倒换

5.1.7.4 设备倒换原因（ESwitchReason_T）

定义	
typedef string ESwitchReason_T;	
说明	
对象说明	设备倒换原因。 "SR_NA"表示未知。 "SR_E_FAILURE"表示设备失效。 "SR_MANUAL"表示人工倒换

5.1.7.5 保护倒换命令类型（ProtectionCommand_T）

定义	

enum ProtectionCommand_T {
PC_CLEAR,
PC_LOCKOUT,
PC_FORCED_SWITCH,
PC_MANUAL_SWITCH,
PC_EXERCISER
};	

说明	
对象说明	保护倒换命令类型
类型取值	取值说明
PC_CLEAR	清除
PC_LOCKOUT	锁定倒换
PC_FORCED_SWITCH	强制倒换
PC_MANUAL_SWITCH	人工倒换
PC_EXERCISER	练习倒换

5.1.7.6 保护组类型（ProtectionGroupType_T）

定义

enum ProtectionGroupType_T {
PGT_MSP_1_PLUS_1,
PGT_MSP_1_FOR_N,
PGT_2_FIBER_BLSR,
PGT_4_FIBER_BLSR
};

说明	
对象说明	保护组类型
类型取值	取值说明
PGT_MSP_1_PLUS_1	1+1 保护
PGT_MSP_1_FOR_N	1：*N* 保护
PGT_2_FIBER_BLSR	二纤环保护
PGT_4_FIBER_BLSR	四纤环保护

5.1.7.7 设备保护组类型（EProtectionGroupType_T）

定义	
typedef string EProtectionGroupType_T;	
说明	
对象说明	设备保护组类型

5.1.7.8 恢复模式（ReversionMode_T）

定义	

enum ReversionMode_T {
RM_UNKNOWN,
RM_NON_REVERTIVE,
RM_REVERTIVE
};	

说明	
对象说明	恢复模式
类型取值	取值说明
RM_UNKNOWN	不清楚
RM_NON_REVERTIVE	不恢复
RM_REVERTIVE	自动恢复

5.1.7.9 保护组（ProtectionGroup_T）

定义	

struct ProtectionGroup_T {
globaldefs::NamingAttributes_T name;
string userLabel;
string nativeEMSName;
string owner;
ProtectionGroupType_T protectionGroupType;
ProtectionSchemeState_T protectionSchemeState;
ReversionMode_T reversionMode;
transmissionParameters::LayerRate_T rate;
globaldefs::NamingAttributesList_T pgpTPList;
globaldefs::NVSList_T pgpParameters;
globaldefs::NVSList_T additionalInfo;
};	

说明	
对象说明	保护组信息
属性名	属性说明

说明	
name	保护组名称
userLabel	保护组的用户标签
nativeEMSName	保护组的本地名称
owner	保护组的所有者
protectionGroupType	保护组类型
protectionSchemeState	保护方案的锁定状态
reversionMode	恢复模式
rate	速率
pgpTPList	构成保护组的 TP 列表，保护 TP 标识和被保护 TP 标识的有序列表
pgpParameters	保护组参数，见本标准第 4 部分 4.7.5 节
additionalInfo	附加信息。 包括： — 子类型(可选)

5.1.7.10 保护组列表（ProtectionGroupList_T）

定义	
typedef sequence <ProtectionGroup_T> ProtectionGroupList_T;	
说明	
对象说明	保护组列表

5.1.7.11 设备保护组（EProtectionGroup_T）(可选)

定义	

struct EProtectionGroup_T {
globaldefs::NamingAttributes_T name;
string userLabel;
string nativeEMSName;
string owner;
EProtectionGroupType_T eProtectionGroupType;
ProtectionSchemeState_T protectionSchemeState;
ReversionMode_T reversionMode;
globaldefs::NamingAttributesList_T protectedList;
globaldefs::NamingAttributesList_T protectingList;
globaldefs::NVSList_T ePgpParameters;
globaldefs::NVSList_T additionalInfo;
};	

说明	
对象说明	设备保护组信息
属性名	属性说明
name	设备保护组名称
userLabel	设备保护组的用户标签
nativeEMSName	设备保护组的本地名称
owner	设备保护组的所有者
eProtectionGroupType	设备保护组类型
protectionSchemeState	保护方案的锁定状态

说明	
reversionMode	恢复模式
protectedList	被保护的设备列表
protectingList	保护其他设备的设备列表
ePgpParameters	设备保护组参数
additionalInfo	附加信息

5.1.7.12 设备保护组列表（EProtectionGroupList_T）

定义	
typedef sequence <EProtectionGroup_T> EProtectionGroupList_T;	
说明	
对象说明	设备保护组列表

5.1.7.13 倒换数据（SwitchData_T）

定义	

struct SwitchData_T {
ProtectionType_T protectionType;
SwitchReason_T switchReason;
transmissionParameters::LayerRate_T layerRate;
globaldefs::NamingAttributes_T groupName;
globaldefs::NamingAttributes_T protectedTP;
globaldefs::NamingAttributes_T switchToTP;
globaldefs::NVSList_T additionalInfo;
};	

说明	
对象说明	倒换数据
属性名	属性说明
protectionType	保护类型
switchReason	倒换原因
layerRate	层速率
groupName	保护组名称
protectedTP	被保护的 TP
switchToTP	倒换到的 TP
additionalInfo	附加信息

5.1.7.14 倒换数据列表（SwitchDataList_T）

定义	
typedef sequence<SwitchData_T> SwitchDataList_T;	
说明	
对象说明	倒换数据列表

5.1.7.15 设备倒换数据（ESwitchData_T）

定义
struct ESwitchData_T { EProtectionGroupType_T eProtectionGroupType; ESwitchReason_T eSwitchReason;

定义
globaldefs::NamingAttributes_T ePGPName; globaldefs::NamingAttributes_T protectedE; globaldefs::NamingAttributes_T switchToE; globaldefs::NVSList_T additionalInfo; };

说明	
对象说明	设备倒换数据
属性名	属性说明
eProtectionGroupType	设备保护类型
eSwitchReason	倒换原因
ePGPName	设备保护组名称
protectedE	被保护的设备
switchToE	倒换到的设备
additionalInfo	附加信息

5.1.7.16 设备倒换数据列表（ESwitchDataList_T）

定义
typedef sequence<ESwitchData_T> ESwitchDataList_T;

说明	
对象说明	设备倒换数据列表

5.1.7.17 路径网络保护组（TrailNtwProtection_T）

定义

struct TrailNtwProtection_T
{
globaldefs::NamingAttributes_T name;
string userLabel;
string nativeEMSName;
string owner;
string protectionType;
protection::ProtectionGroupType_T protectionGroupType;
protection::ProtectionSchemeState_T protectionSchemeState;
protection::ReversionMode_T reversionMode;
transmissionParameters::LayerRate_T rate;
string trailNtwProtectionType;
globaldefs::NamingAttributes_T protectionGroupAName;
globaldefs::NamingAttributes_T protectionGroupZName;
globaldefs::NamingAttributesList_T pgATPList;
globaldefs::NamingAttributesList_T pgZTPList;
globaldefs:: NamingAttributesMultipleList_T workerTrailList;
globaldefs:: NamingAttributesList_T protectionTrail;
globaldefs::NVSList_T tnpParameters;
string apsFunction;
string networkAccessDomain;
globaldefs::NVSList_T additionalInfo;
};

说明	
对象说明	保护组信息
属性名	属性说明
name	TNP 保护组名称
userLabel	TNP 保护组的用户标签
nativeEMSName	TNP 保护组的本地名称
owner	TNP 保护组的所有者
protectionType	保护类型，用于区分 SNCP 保护组和其他类型的保护组
protectionGroupType	保护组类型
protectionSchemeState	保护方案的锁定状态
reversionMode	恢复模式
rate	速率
trailNtwProtectionType	TNP 类型，开环还是闭环
protectionGroupAName	A 端保护组名称
protectionGroupAName	Z 端保护组名称
pgATPList	A 端保护组所含的 TP 名称列表。一般工作 TP 在前，保护 TP 在后
pgZTPList	Z 端保护组所含的 TP 名称列表。一般工作 TP 在前，保护 TP 在后
workTrailList	SNCP 工作路径名称列表
protectionTrail	SNCP 保护路径名称列表
tnpParameters	保护参数。用于描述路径网络保护中的参数信息，例如倒换模式、倒换位置、WTR 等
apsFunction	APS 协议类型，例如："G.783"、"Legacy"、"T-MPLS"
networkAccessDomain	分配给 TNP 的网络接入域
additionalInfo	附加信息

5.1.7.18 路径网络保护组列表（TrailNtwProtectionList_T）

定义	
typedef sequence < TrailNtwProtection_T> TrailNtwProtectionList_T;	
说明	
对象说明	网络保护组列表

5.1.7.19 路径网络保护组创建数据（TrailNtwProtCreateData_T）

定义	

struct TrailNtwProtCreateData_T
{
transmissionParameters::LayerRate_T rate;
string trailNtwProtectionType;
globaldefs::NamingAttributes_T protectionGroupAName;
globaldefs::NamingAttributes_T protectionGroupZName;
TrailNtwProtModifyData_T modifiableAttributes;
};	

说明	
对象说明	保护组信息
属性名	属性说明
rate	速率
trailNtwProtectionType	TNP 类型，开环还是闭环

说明	
protectionGroupAName	A 端保护组名称
protectionGroupZName	Z 端保护组名称
modifiableAttributes	TNP 修改数据

5.1.7.20 路径网络保护组修改数据（TrailNtwProtModifyData_T）

定义
struct TrailNtwProtModifyData _T { string userLabel; string forceUniqueness; string nativeEMSName; string owner; protection::ProtectionGroupType_T protectionGroupType; protection::ReversionMode_T reversionMode; globaldefs::NamingAttributesList_T pgATPList; globaldefs::NamingAttributesList_T pgZTPList; globaldefs::NVSList_T tnpParameters; string apsFunction; string networkAccessDomain; globaldefs::NVSList_T additionalInfo; };

说明	
对象说明	保护组信息
属性名	属性说明
userLabel	TNP 保护组的用户标签
forceUniqueness	用户标签是否唯一标识。若设为是，则当输入的用保护标签所在的 TNP 正在使用时会出错
nativeEMSName	TNP 保护组的本地名称
owner	TNP 保护组的所有者
reversionMode	恢复模式
pgATPList	A 端保护组所含的 TP 名称列表。一般工作 TP 在前，保护 TP 在后
pgZTPList	Z 端保护组所含的 TP 名称列表。一般工作 TP 在前，保护 TP 在后
tnpParameters	保护参数。用于描述路径网络保护组中的参数信息，例如倒换模式、倒换位置、WTR 等
apsFunction	APS 协议类型，例如"G.783"、"Legacy"、"T-MPLS"
networkAccessDomain	分配给 TNP 的网络接入域
additionalInfo	附加信息

5.1.7.21 路径网络保护倒换数据（TNPSwitchData_T）

定义
struct TNPSwitchData_T { globaldefs::NamingAttributes_T tnpName; ProtectionType_T protectionType; SwitchReason_T switchReason; transmissionParameters::LayerRate_T layerRate; globaldefs::NamingAttributesList_T groupNameList; globaldefs:: NamingAttributesMultipleList_T workTrailList;

定义
globaldefs:: NamingAttributesMultipleList_T protectionTrailList; globaldefs::NVSList_T additionalInfo; };

说明	
对象说明	倒换数据
属性名	属性说明
tnpName	TNP 标识
protectionType	保护类型
switchReason	倒换原因
layerRate	层速率
groupNameList	保护组名称列表
workTrailList	工作路径列表
protectionTrailList	保护路径列表
additionalInfo	附加信息

5.1.7.22 ProtectionGroupIterator_I 接口

5.1.7.22.1 从迭代器中查询数据（next_n）

定义
boolean next_n(in unsigned long how_many, out ProtectionGroupList_T pgpList) raises (globaldefs::ProcessingFailureException);

说明	
功能描述	从迭代器中查询 *n* 条记录。当 how_many≥迭代器剩余记录数时，系统在数据取完后应该自动销毁该迭代器对象
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	pgpList：首次查询返回的保护组列表
操作异常	EXCPT_INTERNAL_ERROR

5.1.7.22.2 查询迭代器记录条数（getLength）

定义
unsigned long getLength() raises (globaldefs::ProcessingFailureException);

说明	
功能描述	查询迭代器中数据记录的总条数（注意这里指的是迭代器中数据记录的总条数，该值在迭代器生命周期内是不变的）
输入参数	无
输入/输出参数	无
输出参数	无
返回值	unsigned long：迭代器中包含的总的记录条数
操作异常	EXCPT_INTERNAL_ERROR

5.1.7.22.3 销毁迭代器对象（destroy）

定义	
void destroy() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	销毁迭代器对象
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR

5.1.7.23 EProtectionGroupIterator_I 接口(可选)

5.1.7.23.1 从迭代器中查询数据（next_n）

定义	
boolean next_n(in unsigned long how_many, out EProtectionGroupList_T ePGPList) raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	从迭代器中查询 *n* 条记录。当 how_many≥迭代器剩余记录数时，系统在数据取完后应该自动销毁该迭代器对象
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	ePGPList：首次查询返回的设备保护组列表
操作异常	EXCPT_INTERNAL_ERROR

5.1.7.23.2 查询迭代器记录条数（getLength）

定义	
unsigned long getLength() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	查询迭代器中数据记录的总条数（注意这里指的是迭代器中数据记录的总条数，该值在迭代器生命周期内是不变的）
输入参数	无
输入/输出参数	无
输出参数	无
返回值	unsigned long：迭代器中包含的总的记录条数
操作异常	EXCPT_INTERNAL_ERROR

5.1.7.23.3 销毁迭代器对象（destroy）

定义
void destroy() raises (globaldefs::ProcessingFailureException);

说明	
功能描述	销毁迭代器对象
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR

5.1.7.24 TrailNtwProtectionIterator_I 接口

5.1.7.24.1 从迭代器中查询数据（next_n）

定义	
boolean next_n(in unsigned long how_many, out TrailNtwProtectionList_T tnpList) raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	从迭代器中查询 *n* 条记录。当 how_many≥迭代器剩余记录数时，系统在数据取完后应该自动销毁该迭代器对象
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	tnpList：首次查询返回的网络保护组列表
操作异常	EXCPT_INTERNAL_ERROR

5.1.7.24.2 查询迭代器记录条数（getLength）

定义	
unsigned long getLength() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	查询迭代器中数据记录的总条数（注意这里指的是迭代器中数据记录的总条数，该值在迭代器生命周期内是不变的）
输入参数	无
输入/输出参数	无
输出参数	无
返回值	unsigned long：迭代器中包含的总的记录条数
操作异常	EXCPT_INTERNAL_ERROR

5.1.7.24.3 销毁迭代器对象（destroy）

定义	
void destroy() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	销毁迭代器对象
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR

5.1.7.25 ProtectionMgr_I 接口（从 common::Common_I 继承）

5.1.7.25.1 查询网元下所有保护组信息（getAllProtectionGroups）

定义	
void getAllProtectionGroups(in globaldefs::NamingAttributes_T meName, in unsigned long how_many, out ProtectionGroupList_T pgList, out ProtectionGroupIterator_I pgpIt) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询网元下所有保护组信息
输入参数	meName：网元名称。 how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	pgList：保护组列表。 pgpIt：迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.7.25.2 查询网元下所有设备保护组信息（getAllEProtectionGroups）

定义	
void getAllEProtectionGroups(in globaldefs::NamingAttributes_T meName, in unsigned long how_many, out EProtectionGroupList_T epgpList, out EProtectionGroupIterator_I epgpIt) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询网元下所有设备保护组信息
输入参数	meName：网元名称。 how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	epgpList：设备保护组列表。 epgpIt：迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.7.25.3 查询指定的保护组信息（getProtectionGroup）

定义	
void getProtectionGroup(in globaldefs::NamingAttributes_T pgName, out protection::ProtectionGroup_T protectionGroup) raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	查询指定的保护组信息
输入参数	pgName：保护组名称
输入/输出参数	无
输出参数	protectionGroup：保护组
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

5.1.7.25.4 查询指定的设备保护组信息（getEProtectionGroup）

定义	
void getEProtectionGroup(in globaldefs::NamingAttributes_T ePGPname, out protection::EProtectionGroup_T eProtectionGroup) raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	查询指定的设备保护组信息
输入参数	pgName：保护组名称
输入/输出参数	无
输出参数	protectionGroup：保护组
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

5.1.7.25.5 查询所有的保护功能已被封锁的不可抢占的通道 TP 名称（getAllNUTTPNames）(可选)

定义	
void getAllNUTTPNames(in globaldefs::NamingAttributes_T pgName, in unsigned long how_many, out globaldefs::NamingAttributesList_T nameList, out globaldefs::NamingAttributesIterator_I nameIt) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询所有的保护功能已被封锁的不可抢占的通道 TP 名称
输入参数	pgName：保护组名称。 how_many：首次迭代查询返回的数据数目
输入/输出参数	无

说明	
输出参数	nameList：名称列表。 nameIt：迭代器
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.7.25.6 查询保护组中可配置为抢占式的额外路径的终端点（getAllPreemptibleTPNames）(可选)

定义
void getAllPreemptibleTPNames(in globaldefs::NamingAttributes_T pgName, in unsigned long how_many, out globaldefs::NamingAttributesList_T nameList, out globaldefs::NamingAttributesIterator_I nameIt) raises(globaldefs::ProcessingFailureException);

说明	
功能描述	查询保护组中可配置为抢占式的额外路径的终端点
输入参数	pgName：保护组名称。 how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	nameList：名称列表。 nameIt：迭代器
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.7.25.7 查询所有被保护的 TP 名称（getAllProtectedTPNames）(可选)

定义
void getAllProtectedTPNames(in globaldefs::NamingAttributes_T pgName, in unsigned long how_many, out globaldefs::NamingAttributesList_T nameList, out globaldefs::NamingAttributesIterator_I nameIt) raises(globaldefs::ProcessingFailureException);

说明	
功能描述	查询所有被保护的 TP 名称
输入参数	pgName：保护组名称。 how_many：首次迭代查询返回的数据数目
输入/输出参数	无

说明	
输出参数	nameList：名称列表。 nameIt：迭代器
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.7.25.8 查询保护倒换数据（retrieveSwitchData）

定义	
void retrieveSwitchData(in globaldefs::NamingAttributes_T reliableSinkCtpOrGroupName, out protection::SwitchDataList_T switchData) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	指定对象名称，查询该对象的保护倒换数据
输入参数	reliableSinkCtpOrGroupName：保护组名称或宿 CTP 名称
输入/输出参数	无
输出参数	switchData：保护倒换数据列表
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

5.1.7.25.9 查询设备保护倒换数据（retrieveESwitchData）

定义	
void retrieveESwitchData(in globaldefs::NamingAttributes_T ePGPName, out protection::ESwitchDataList_T eSwitchDataList) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	指定对象名称，查询该对象的设备保护倒换数据
输入参数	ePGPName：设备保护组名称
输入/输出参数	无
输出参数	eSwitchDataList：设备保护倒换数据列表
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

5.1.7.25.10 执行保护倒换命令（performProtectionCommand）

定义	
void performProtectionCommand(in ProtectionCommand_T protectionCommand, in globaldefs::NamingAttributes_T reliableSinkCtpOrGroupName, in globaldefs::NamingAttributes_T fromTp, in globaldefs::NamingAttributes_T toTp, out protection::SwitchData_T switchData) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	执行保护倒换命令
输入参数	protectionCommand：保护倒换命令。 reliableSinkCtpOrGroupName：保护组名称或宿 CTP 名称。 fromTp：被倒换的对象名称。 toTp：倒换后到达的对象名称
输入/输出参数	无
输出参数	switchData：倒换后的保护倒换状态
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.1.7.25.11 查询所有路径网络保护组（getAllTrailNtwProtections）

定义	
void getAllTrailNtwProtections (in globaldefs::NamingAttributes_T resourceName, in unsigned long how_many, out TrailNtwProtectionList_T tnpList, out TrailNtwProtectionIterator_I tnpIt) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询顶层子网下的所有网络保护组信息
输入参数	subnetName：资源对象名称，对象可以为 ME，PG 或 SNC。 how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	tnpList：网络保护组列表。 tnpIt：迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.7.25.12 创建路径网络保护（createTrailNtwProtection）

定义	
void createTrailNtwProtection (in TrailNtwProtCreateData_T tnpCreateData, out TrailNtwProtection_T theTNP, out string errorReason) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	创建路径网络保护组
输入参数	tnpCreateData：待创建的路径网络保护信息
输入/输出参数	无
输出参数	theTNP：创建后的路径网络保护组。 errorReason：错误原因
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.7.25.13 删除路径网络保护（deleteTrailNtwProtection）

定义	
void deleteTrailNtwProtection(in globaldefs::NamingAttributes_T tnpName, in boolean keepPGs, in globaldefs::NamingAttributes_T swapTPname, inout globaldefs::NVSList_T additionalInfo, out protection::ProtectionGroupList_T deletedPGList, out TrailNtwProtection_T deletedTNP, out string errorReason) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	删除指定的路径网络保护组
输入参数	tnpName：指定的待删除的路径网络保护组名称。 keepPGs：包含的 PG 组是否删除标识。是，则包含的 PG 保留；否，则包含的 PG 删除。 swapTPname：客户侧流量是否交替模式。若是 SWAP 模式，则之前属于工作路径的流量将被保护路径支持；若不是，客户侧流量将还是由工作路径支持
输入/输出参数	additionalInfo：附加信息
输出参数	deletedPGList：被删除的保护组信息列表。 deletedTNP：被删除的网络路径保护组信息。 errorReason：返回的错误原因
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.7.25.14 修改路径网络保护组（modifyTrailNtwProtection）

定义	
void modifyTrailNtwProtection(in globaldefs::NamingAttributes_T tnpName, in TrailNtwProtModifyData_T tnpModifyData, out TrailNtwProtection_T modifiedTNP, out string errorReason) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	修改指定的网络保护组
输入参数	tnpName：指定的待修改的 TNP 名称。 tnpModifyData：TNP 修改数据信息
输入/输出参数	无
输出参数	modifiedTNP：修改后的路径网络保护组信息。 errorReason：错误原因
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.7.25.15 执行保护倒换命令（performTnpProtectionCommand）

定义	
void performTnpProtectionCommand(in protection::ProtectionCommand_T protectionCommand, in TNetworkProtectionGroup_T oneTnpData, in globaldefs::NamingAttributes_T reliableSinkCtpOrGroupName, in globaldefs::NamingAttributes_T fromTp, in globaldefs::NamingAttributes_T toTp, out TNPSwitchData_T tnpSwitchData) raises (globaldefs::ProcessingFailureException); };	
说明	
功能描述	执行 TNP 保护倒换命令
输入参数	protectionCommand：保护倒换命令。 oneTnpData：指定的路径网络保护组。 reliableSinkCtpOrGroupName：保护组名称或宿 CTP 名称。 fromTp：被倒换的对象名称。 toTp：倒换后到达的对象名称
输入/输出参数	无
输出参数	switchData：倒换后的保护倒换数据。倒换成功后会收到倒换通知，结构为 switchData
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.1.8 子网模块（module multiLayerSubnetwork）

说明：对于端到端业务的管理，可以采用 SNC 或 FDFr 中任意一种数据格式表达，不根据业务类型进行区分。

5.1.8.1 子网的拓扑类型（Topology_T）

定义

enum Topology_T
{
TOPO_SINGLETON,
TOPO_CHAIN,
TOPO_PSR,
TOPO_OPEN_PSR,
TOPO_SPRING,
TOPO_OPEN_SPRING,
TOPO_MESH
};

说明	
对象说明	子网的拓扑类型
类型取值	取值说明
TOPO_SINGLETON	简单节点
TOPO_CHAIN	线型
TOPO_PSR	通道倒换环
TOPO_OPEN_PSR	开放的通道倒换环
TOPO_SPRING	共享环
TOPO_OPEN_SPRING	开放的共享环
TOPO_MESH	网孔型

5.1.8.2 EMS 的操作自由度（EMSFreedomLevel_T）

定义

enum EMSFreedomLevel_T {
EMSFL_CC_AT_SNC_LAYER,
EMSFL_TERMINATE_AND_MAP,
EMSFL_HIGHER_ORDER_SNCS,
EMSFL_RECONFIGURATION
};

说明	
对象说明	EMS 的操作自由度
类型取值	取值说明
EMSFL_CC_AT_SNC_LAYER	EMS 可以创建或删除可被和已被 SNC 使用的交叉连接
EMSFL_TERMINATE_AND_MAP	在 EMSFL_CC_AT_SNC_LAYER 基础上，EMS 可以映射去映射或通道化去通道化可被和已被 SNC 使用的终端点
EMSFL_HIGHER_ORDER_SNCS	在 EMSFL_TERMINATE_AND_MAP 基础上，EMS 可以创建或删除可被或已被用作承载 SNC 的高阶交叉连接
EMSFL_RECONFIGURATION	EMS 可以进行任何操作，即 NMS 不限制 EMS 进行与 SNC 相关的操作

5.1.8.3 多层子网（MultiLayerSubnetwork_T）

定义

struct MultiLayerSubnetwork_T {
globaldefs::NamingAttributes_T name;
string userLabel;
string nativeEMSName;
string owner;
Topology_T subnetworkType;
transmissionParameters::LayerRateList_T supportedRates;
globaldefs::NVSList_T additionalInfo;
};

说明	
对象说明	多层子网信息
属性名	属性说明
name	多层子网名称
userLabel	多层子网的用户标签
nativeEMSName	多层子网的本地名称
owner	多层子网的所有者
subnetworkType	多层子网类型
supportedRates	多层子网支持的业务速率
additionalInfo	附加信息

5.1.8.4 多层子网列表（SubnetworkList_T）

定义
typedef sequence<MultiLayerSubnetwork_T> SubnetworkList_T;

说明	
对象说明	多层子网列表

5.1.8.5 SubnetworkIterator_I 接口

5.1.8.5.1 从迭代器中查询数据（next_n）

定义
boolean next_n(in unsigned long how_many, out SubnetworkList_T subnetworkList) raises (globaldefs::ProcessingFailureException);

说明	
功能描述	从迭代器中查询 *n* 条记录。当 how_many≥迭代器剩余记录数时，系统在数据取完后应该自动销毁该迭代器对象
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	subnetworkList：首次查询返回的多层子网列表
操作异常	EXCPT_INTERNAL_ERROR

5.1.8.5.2 查询迭代器记录条数（getLength）

定义	
unsigned long getLength() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	查询迭代器中数据记录的总条数（注意这里指的是迭代器中数据记录的总条数，该值在迭代器生命周期内是不变的）
输入参数	无
输入/输出参数	无
输出参数	无
返回值	unsigned long：迭代器中包含的总的记录条数
操作异常	EXCPT_INTERNAL_ERROR

5.1.8.5.3 销毁迭代器对象（destroy）

定义	
void destroy() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	销毁迭代器对象
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR

5.1.8.6 MultiLayerSubnetworkMgr_I 接口（从 common::Common_I 继承）

5.1.8.6.1 查询子网下所有的网元信息（getAllManagedElements）

定义	
void getAllManagedElements(in globaldefs::NamingAttributes_T subnetName, in unsigned long how_many, out managedElement::ManagedElementList_T meList, out managedElement::ManagedElementIterator_I meIt) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询子网下所有的网元信息
输入参数	subnetName：子网名称。 how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	meList：网元列表。 meIt：迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.8.6.2 查询子网下所有的网元名称（getAllManagedElementNames）

定义	
void getAllManagedElementNames(in globaldefs::NamingAttributes_T subnetName, in unsigned long how_many, out globaldefs::NamingAttributesList_T nameList, out globaldefs::NamingAttributesIterator_I nameIt) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询子网下所有的网元名称
输入参数	subnetName：子网名称。 how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	nameList：名称列表。 nameIt：迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.8.6.3 查询多层子网（getMultiLayerSubnetwork）

定义	
void getMultiLayerSubnetwork(in globaldefs::NamingAttributes_T subnetName, out MultiLayerSubnetwork_T subnetwork) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询多层子网
输入参数	subnetName：子网名称
输入/输出参数	无
输出参数	subnetwork：多层子网信息
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

5.1.8.6.4 查询子网下所有的拓扑连接（getAllTopologicalLinks）

定义
void getAllTopologicalLinks(in globaldefs::NamingAttributes_T subnetName, in unsigned long how_many, out topologicalLink::TopologicalLinkList_T topoList, out topologicalLink::TopologicalLinkIterator_I topoIt) raises(globaldefs::ProcessingFailureException);

说明	
功能描述	查询子网下所有的拓扑连接
输入参数	subnetName：子网名称。 how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	topoList：拓扑连接列表。 topoIt：迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.8.6.5 查询子网下所有的拓扑连接名称（getAllTopologicalLinkNames）

定义
void getAllTopologicalLinkNames(in globaldefs::NamingAttributes_T subnetName, in unsigned long how_many, out globaldefs::NamingAttributesList_T nameList, out globaldefs::NamingAttributesIterator_I nameIt) raises(globaldefs::ProcessingFailureException);

说明	
功能描述	查询子网下所有的拓扑连接名称
输入参数	subnetName：子网名称。 how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	nameList：名称列表。 nameIt：迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.8.6.6 查询指定的拓扑连接（getTopologicalLink）

定义
void getTopologicalLink(in globaldefs::NamingAttributes_T topoLinkName, out topologicalLink::TopologicalLink_T topoLink) raises(globaldefs::ProcessingFailureException);

说明	
功能描述	查询拓扑连接
输入参数	topoLinkName：拓扑连接名称

说明	
输入/输出参数	无
输出参数	topoLink：拓扑连接信息
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

5.1.8.6.7 查询子网下所有的边界点（getAllEdgePoints）（可选）

定义
void getAllEdgePoints(in globaldefs::NamingAttributes_T subnetName, in transmissionParameters::LayerRateList_T tpLayerRateList, in transmissionParameters::LayerRateList_T connectionLayerRateList, in unsigned long how_many, out terminationPoint::TerminationPointList_T tpList, out terminationPoint::TerminationPointIterator_I tpIt) raises(globaldefs::ProcessingFailureException);

说明	
功能描述	查询子网下所有的边界点
输入参数	subnetName：子网名称。 tpLayerRateList：TP 点包含的速率参数列表。 connectionLayerRateList：连接速率列表。 how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	tpList：TP 列表。 tpIt：迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.8.6.8 查询子网下所有的边界点名称（getAllEdgePointNames）（可选）

定义
void getAllEdgePointNames(in globaldefs::NamingAttributes_T subnetName, in transmissionParameters::LayerRateList_T layerRateList, in transmissionParameters::LayerRateList_T connectionLayerRateList, in unsigned long how_many, out globaldefs::NamingAttributesList_T nameList, out globaldefs::NamingAttributesIterator_I nameIt) raises(globaldefs::ProcessingFailureException);

说明	
功能描述	查询子网下所有的边界点名称
输入参数	subnetName：子网名称。 tpLayerRateList：TP 点包含的速率参数列表。 connectionLayerRateList：连接速率列表。 how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	nameList：名称列表。 nameIt：迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.8.6.9 查询保护相关 TP（getAssociatedTP）(可选)

定义
void getAssociatedTP(in globaldefs::NamingAttributes_T tpName, out terminationPoint::TerminationPointList_T tpList) raises(globaldefs::ProcessingFailureException);

说明	
功能描述	查询保护相关 TP
输入参数	tpName：TP 名称
输入/输出参数	无
输出参数	tpList：TP 列表
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

5.1.8.6.10 查询所有的 SNC（getAllSubnetworkConnections）

定义
void getAllSubnetworkConnections(in globaldefs::NamingAttributes_T subnetName, in transmissionParameters::LayerRateList_T connectionRateList, in unsigned long how_many, out subnetworkConnection::SubnetworkConnectionList_T sncList, out subnetworkConnection::SNCIterator_I sncIt) raises(globaldefs::ProcessingFailureException);

说明	
功能描述	查询所有的 SNC
输入参数	subnetName：子网名称。 connectionLayerRateList：连接速率列表。 how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	sncList：SNC 列表。 sncIt：SNC 迭代查询接口
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.8.6.11 查询所有的 SNC 名称（getAllSubnetworkConnectionNames）

定义
void getAllSubnetworkConnectionNames(in globaldefs::NamingAttributes_T subnetName, in transmissionParameters::LayerRateList_T connectionRateList, in unsigned long how_many, out globaldefs::NamingAttributesList_T nameList, out globaldefs::NamingAttributesIterator_I nameIt) raises(globaldefs::ProcessingFailureException);

说明	
功能描述	查询所有的 SNC 名称
输入参数	subnetName：子网名称。 connectionLayerRateList：连接速率列表。 how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	nameList：名称列表。 nameIt：名称迭代查询接口
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.8.6.12 查询指定 TP 下的所有 SNC（getAllSubnetworkConnectionsWithTP）

定义
void getAllSubnetworkConnectionsWithTP (in globaldefs::NamingAttributes_T tpName, in transmissionParameters::LayerRateList_T connectionRateList, in unsigned long how_many,

定义	
out subnetworkConnection::SubnetworkConnectionList_T sncList, out subnetworkConnection::SNCIterator_I sncIt) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询指定 TP 下的所有 SNC
输入参数	tpName：TP 名称。 connectionLayerRateList：连接速率列表。 how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	sncList：SNC 列表。 sncIt：SNC 迭代查询接口
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.8.6.13 查询 SNC 路由(getRoute）

定义	
void getRoute(in globaldefs::NamingAttributes_T sncName, in boolean includeHigherOrderCCs, out subnetworkConnection::Route_T route) raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	指定 SNC 名称，查询 SNC 路由信息
输入参数	sncName：SNC 名称； includeHigherOrderCCs：是否返回承载 SNC 的路由的高阶交叉连接
输入/输出参数	无
输出参数	route：SNC 的路由
操作异常	EXCPT_INTERNAL_ERROR EXCPT_NOT_IMPLEMENTED EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY

5.1.8.6.14 查询指定 TP 下的所有 SNC 名称（getAllSubnetworkConnectionNamesWithTP）

定义
void getAllSubnetworkConnectionNamesWithTP (in globaldefs::NamingAttributes_T tpName, in transmissionParameters::LayerRateList_T connectionRateList, in unsigned long how_many, out globaldefs::NamingAttributesList_T nameList,

定义	
out globaldefs::NamingAttributesIterator_I nameIt) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询指定 TP 下的所有 SNC 名称
输入参数	tpName：TP 名称。 connectionLayerRateList：连接速率列表。 how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	nameList：名称列表。 nameIt：名称迭代查询接口
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.8.6.15 查询指定的 SNC（getSNC）

定义	
void getSNC(in globaldefs::NamingAttributes_T sncName, out subnetworkConnection::SubnetworkConnection_T snc) raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	查询指定的 SNC
输入参数	sncName：SNC 名称
输入/输出参数	无
输出参数	snc：SNC 信息
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

5.1.8.6.16 通过用户标签查询 SNC（getSNCsByUserLabel）（可选）

定义	
void getSNCsByUserLabel(in string userLabel, out subnetworkConnection::SubnetworkConnectionList_T sncList) raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	通过用户标签查询 SNC
输入参数	userLabel：子网用户标签
输入/输出参数	无
输出参数	sncList：SNC 列表

说明	
操作异常	EXCPT_INTERNAL_ERROR EXCPT_UNABLE_TO_COMPLY EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

5.1.8.6.17 创建 SNC（createSNC）

定义
void createSNC(in subnetworkConnection::SNCCreateData_T createData, in subnetworkConnection::GradesOfImpact_T tolerableImpact, in EMSFreedomLevel_T emsFreedomLevel, out subnetworkConnection::SubnetworkConnection_T theSNC, out string errorReason) raises (globaldefs::ProcessingFailureException);

说明	
功能描述	创建 SNC
输入参数	createData：子网连接创建数据。 tolerableImpact：可容忍的业务影响级别。 emsFreedomLevel：EMS 操作自由度级别
输入/输出参数	无
输出参数	theSNC：创建成功的 SNC。 errorReason：错误原因
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_PROTECTION_EFFORT_NOT_MET EXCPT_UNABLE_TO_COMPLY EXCPT_UNSUPPORTED_ROUTING_CONSTRAINTS EXCPT_USERLABEL_IN_USE EXCPT_OBJECT_IN_USE

5.1.8.6.18 激活 SNC（activateSNC）

定义
void activateSNC(in globaldefs::NamingAttributes_T sncName, in subnetworkConnection::GradesOfImpact_T tolerableImpact, in EMSFreedomLevel_T emsFreedomLevel, inout subnetworkConnection::TPDataList_T tpsToModify, out subnetworkConnection::SubnetworkConnection_T theSNC, out string errorReason) raises (globaldefs::ProcessingFailureException);

说明	
功能描述	激活 SNC
输入参数	sncName：SNC 名称。 tolerableImpact：可容忍的业务影响级别。 emsFreedomLevel：EMS 操作自由度级别
输入/输出参数	tpsToModify：修改的 TP 列表
输出参数	theSNC：激活后的 SNC。 errorReason：错误原因
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_PROTECTION_EFFORT_NOT_MET EXCPT_UNABLE_TO_COMPLY EXCPT_UNSUPPORTED_ROUTING_CONSTRAINTS EXCPT_USERLABEL_IN_USE EXCPT_OBJECT_IN_USE EXCPT_NE_COMM_LOSS EXCPT_NOT_IN_VALID_STATE

5.1.8.6.19 创建并激活 SNC（createAndActivateSNC）

定义
void createAndActivateSNC(in subnetworkConnection::SNCCreateData_T createData, in subnetworkConnection::GradesOfImpact_T tolerableImpact, in EMSFreedomLevel_T emsFreedomLevel, inout subnetworkConnection::TPDataList_T tpsToModify, out subnetworkConnection::SubnetworkConnection_T theSNC, out string errorReason) raises (globaldefs::ProcessingFailureException);

说明	
功能描述	创建并激活 SNC
输入参数	createData：子网连接创建数据。 tolerableImpact：可容忍的业务影响级别。 emsFreedomLevel：EMS 操作自由度级别
输入/输出参数	tpsToModify：修改的 TP 列表
输出参数	theSNC：创建并激活后的 SNC。 errorReason：错误原因
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_PROTECTION_EFFORT_NOT_MET EXCPT_UNABLE_TO_COMPLY EXCPT_UNSUPPORTED_ROUTING_CONSTRAINTS

说明	
操作异常	EXCPT_USERLABEL_IN_USE EXCPT_OBJECT_IN_USE EXCPT_NE_COMM_LOSS EXCPT_NOT_IN_VALID_STATE

5.1.8.6.20 去激活 SNC（deactivateSNC）

定义
void deactivateSNC(in globaldefs::NamingAttributes_T sncName, in subnetworkConnection::GradesOfImpact_T tolerableImpact, in EMSFreedomLevel_T emsFreedomLevel, inout subnetworkConnection::TPDataList_T tpsToModify, out subnetworkConnection::SubnetworkConnection_T theSNC, out string errorReason) raises (globaldefs::ProcessingFailureException);

说明	
功能描述	去激活 SNC
输入参数	sncName：SNC 名称。 tolerableImpact：可容忍的业务影响级别。 emsFreedomLevel：EMS 操作自由度级别
输入/输出参数	tpsToModify：修改的 TP 列表
输出参数	theSNC：去激活后的 SNC。 errorReason：错误原因
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_PROTECTION_EFFORT_NOT_MET EXCPT_UNABLE_TO_COMPLY EXCPT_UNSUPPORTED_ROUTING_CONSTRAINTS EXCPT_USERLABEL_IN_USE EXCPT_OBJECT_IN_USE EXCPT_NE_COMM_LOSS EXCPT_NOT_IN_VALID_STATE

5.1.8.6.21 删除 SNC（deleteSNC）

定义
void deleteSNC(in globaldefs::NamingAttributes_T sncName, in EMSFreedomLevel_T emsFreedomLevel) raises (globaldefs::ProcessingFailureException);

说明	
功能描述	删除 SNC
输入参数	sncName：SNC 名称。 emsFreedomLevel：EMS 操作自由度级别
输入/输出参数	无
输出参数	无
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NOT_IN_VALID_STATE EXCPT_UNABLE_TO_COMPLY

5.1.8.6.22 去激活并删除 SNC（deactivateAndDeleteSNC）

```
定义
void deactivateAndDeleteSNC(
        in    globaldefs::NamingAttributes_T                sncName,
        in    subnetworkConnection::GradesOfImpact_T        tolerableImpact,
        in    EMSFreedomLevel_T                             emsFreedomLevel,
        inout subnetworkConnection::TPDataList_T            tpsToModify,
        out   subnetworkConnection::SubnetworkConnection_T theSNC,
        out   string                                        errorReason)
    raises (globaldefs::ProcessingFailureException);
```

说明	
功能描述	去激活并删除 SNC
输入参数	sncName：SNC 名称。 tolerableImpact：可容忍的业务影响级别。 emsFreedomLevel：EMS 操作自由度级别
输入/输出参数	tpsToModify：修改的 TP 列表
输出参数	theSNC：去激活并删除的 SNC。 errorReason：错误原因
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.1.8.6.23 检查 SNC 创建数据是否有效（checkValidSNC）(可选)

```
定义
void checkValidSNC(
        in subnetworkConnection::SNCCreateData_T createData,
        in subnetworkConnection::TPDataList_T tpsToModify,
        in boolean considerResources,
        out boolean valid)
    raises (globaldefs::ProcessingFailureException);
```

说明	
功能描述	检查 SNC 创建数据是否有效
输入参数	createData：SNC 创建数据。 tpsToModify：TP 修改数据。 considerResources：是否考虑资源分配
输入/输出参数	tpsToModify：修改的 TP 列表
输出参数	valid：是否有效
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_OBJECT_IN_USE EXCPT_INVALID_INPUT EXCPT_NE_COMM_LOSS EXCPT_UNSUPPORTED_ROUTING_CONSTRAINTS EXCPT_USERLABEL_IN_USE EXCPT_UNABLE_TO_COMPLY EXCPT_OBJECT_IN_USE

5.1.8.6.24 修改 SNC（modifySNC）

定义
void modifySNC(in globaldefs::NamingAttributes_T sncName, in string routeId, in subnetworkConnection::SNCModifyData_T SNCModifyData, in subnetworkConnection::GradesOfImpact_T tolerableImpact, in subnetworkConnection::ProtectionEffort_T tolerableImpactEffort, in EMSFreedomLevel_T emsFreedomLevel, inout subnetworkConnection::TPDataList_T tpsToModify, out subnetworkConnection::SubnetworkConnection_T newSNC, out string errorReason) raises (globaldefs::ProcessingFailureException);

说明	
功能描述	修改 SNC
输入参数	sncName：SNC 名称。 routeId：路由标识。 SNCModifyData：SNC 修改数据。 tolerableImpact：可容忍的业务影响级别。 tolerableImpactEffort：可容忍的保护尽力程度。 emsFreedomLevel：EMS 操作自由度级别
输入/输出参数	tpsToModify：修改的 TP 列表
输出参数	newSNC：修改后的 SNC。 errorReason：错误原因
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_OBJECT_IN_USE EXCPT_ENTITY_NOT_FOUND

说明	
操作异常	EXCPT_UNSUPPORTED_ROUTING_CONSTRAINTS EXCPT_USERLABEL_IN_USE EXCPT_UNABLE_TO_COMPLY EXCPT_OBJECT_IN_USE

5.1.8.6.25 查询 SNC 路由(getEXRoute）

定义
void getEXRoute(in globaldefs::NamingAttributes_T sncName, in boolean includeHigherOrderCCs, out EXRoute_T route) raises (globaldefs::ProcessingFailureException);

说明	
功能描述	指定 SNC 名称，查询 SNC 路由信息
输入参数	sncName：SNC 名称； includeHigherOrderCCs：是否返回承载 SNC 的路由的高阶交叉连接
输入/输出参数	无
输出参数	route：SNC 的路由
操作异常	EXCPT_INTERNAL_ERROR EXCPT_NOT_IMPLEMENTED EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY

5.1.9 子网连接模块（module subnetworkConnection）

说明：对于端到端业务的管理，可以采用 SNC 或 FDFr 中任意一种数据格式表达，不根据业务类型进行区分。

5.1.9.1 静态保护级别（StaticProtectionLevel_T）

定义

enum StaticProtectionLevel_T {
PREEMPTIBLE,
UNPROTECTED,
PARTIALLY_PROTECTED,
FULLY_PROTECTED,
HIGHLY_PROTECTED
};

说明	
对象说明	静态保护级别
类型取值	取值说明
PREEMPTIBLE	可抢占
UNPROTECTED	无保护
PARTIALLY_PROTECTED	部分保护
FULLY_PROTECTED	完全保护
HIGHLY_PROTECTED	高级保护

5.1.9.2 保护尽力程度（ProtectionEffort_T）

定义	

enum ProtectionEffort_T
{
EFFORT_WHATEVER,
EFFORT_SAME_OR_BETTER,
EFFORT_SAME_OR_WORSE,
EFFORT_SAME
};	

说明	
对象说明	保护尽力程度
类型取值	取值说明
EFFORT_WHATEVER	其他保护级别也可以
EFFORT_SAME_OR_BETTER	可选择相同或更好的保护级别
EFFORT_SAME_OR_WORSE	可选择相同或差的保护级别
EFFORT_SAME	选择相同的保护级别

5.1.9.3 子网连接状态（SNCState_T）

定义	

enum SNCState_T {
SNCS_NONEXISTENT,
SNCS_PENDING,
SNCS_ACTIVE,
SNCS_PARTIAL
};	

说明	
对象说明	子网连接状态
类型取值	取值说明
SNCS_NONEXISTENT	不存在
SNCS_PENDING	挂起
SNCS_ACTIVE	激活
SNCS_PARTIAL	部分激活

5.1.9.4 业务影响级别（GradesOfImpact_T）

定义	

enum GradesOfImpact_T {
GOI_HITLESS,
GOI_MINOR_IMPACT,
GOI_MAJOR_IMPACT
};	

说明	
对象说明	业务影响级别
类型取值	取值说明

说明	
GOI_HITLESS	无影响
GOI_MINOR_IMPACT	次要影响
GOI_MAJOR_IMPACT	主要影响

5.1.9.5 TP 数据（TPData_T）

定义

struct TPData_T
{
globaldefs::NamingAttributes_T tpName;
terminationPoint::TerminationMode_T tpMappingMode;
transmissionParameters::LayeredParameterList_T transmissionParams;
globaldefs::NamingAttributes_T ingressTrafficDescriptorName;
globaldefs::NamingAttributes_T egressTrafficDescriptorName;
};

说明	
对象说明	终端点数据
类型取值	取值说明
tpName	TP 名称
tpMappingMode	TP 映射方式
transmissionParams	层参数列表
ingressTrafficDescriptorName	入口业务描述符名称
egressTrafficDescriptorName	出口业务描述符名称

5.1.9.6 TP 数据列表（TPDataList_T）

定义
typedef sequence<TPData_T> TPDataList_T;

说明	
对象说明	TP 数据列表

5.1.9.7 SNC 类型（SNCType_T）

定义

enum SNCType_T {
ST_SIMPLE,
ST_ADD_DROP_A,
ST_ADD_DROP_Z,
ST_INTERCONNECT,
ST_DOUBLE_INTERCONNECT,
ST_DOUBLE_ADD_DROP,
ST_OPEN_ADD_DROP,
ST_EXPLICIT
};

说明	
对象说明	SNC 类型
类型取值	取值说明

说明	
ST_SIMPLE	简单型
ST_ADD_DROP_A	A 点 AddDrop 型
ST_ADD_DROP_Z	Z 点 AddDrop 型
ST_INTERCONNECT	互连型
ST_DOUBLE_INTERCONNECT	双互连型
ST_DOUBLE_ADD_DROP	双向 AddDrop 型
ST_OPEN_ADD_DROP	开放的 AddDrop 型
ST_EXPLICIT	其他型

5.1.9.8 重路由允许类型（Reroute_T）

定义	

enum Reroute_T {
RR_NA,
RR_NO,
RR_YES
};	

说明	
对象说明	重路由允许类型
类型取值	取值说明
RR_NA	无关
RR_NO	不允许
RR_YES	允许

5.1.9.9 网络层重路由允许类型（NetworkRouted_T）

定义	

enum NetworkRouted_T{
NR_NA,
NR_NO,
NR_YES
};	

说明	
对象说明	网络层重路由允许类型
类型取值	取值说明
NR_NA	无关
NR_NO	不允许
NR_YES	允许

5.1.9.10 SNC（SubnetworkConnection_T）

定义
struct SubnetworkConnection_T { globaldefs::NamingAttributes_T name; string userLabel; string nativeEMSName; string owner; SNCState_T sncState;

定义
globaldefs::ConnectionDirection_T direction; transmissionParameters::LayerRate_T rate; StaticProtectionLevel_T staticProtectionLevel; SNCType_T sncType; TPDataList_T aEnd; TPDataList_T zEnd; Reroute_T rerouteAllowed; NetworkRouted_T networkRouted; globaldefs::NVSList_T additionalInfo; };

说明	
对象说明	SNC 信息
属性名	属性说明
name	SNC 名称
userLabel	SNC 的用户标签
nativeEMSName	SNC 的本地名称
owner	SNC 的所有者
sncState	SNC 的状态
direction	SNC 的方向
rate	SNC 的速率
staticProtectionLevel	SNC 的静态保护等级
sncType	SNC 类型
aEnd	SNC 的 A 点名称列表
zEnd	SNC 的 Z 点名称列表
rerouteAllowed	重路由允许
networkRouted	网络层重路由允许
additionalInfo	附加信息

5.1.9.11 SNC 列表（SubnetworkConnectionList_T）

定义
typedef sequence<SubnetworkConnection_T> SubnetworkConnectionList_T;

说明	
对象说明	SNC 列表

5.1.9.12 交叉连接（CrossConnect_T）

定义

struct CrossConnect_T {
boolean active;
globaldefs::ConnectionDirection_T direction;
SNCType_T ccType;
globaldefs::NamingAttributesList_T aEndNameList;
globaldefs::NamingAttributesList_T zEndNameList;
globaldefs::NVSList_T additionalInfo;
};

说明	
对象说明	交叉连接信息
属性名	属性说明
active	交叉连接的激活状态
direction	交叉连接的方向
ccType	交叉连接的类型
aEndNameList	交叉连接的 A 点名称列表
zEndNameList	交叉连接的 Z 点名称列表
additionalInfo	附加信息。 包括： —网元标识符； —交叉连接标识符（可选）

5.1.9.13 交叉连接列表（CrossConnectList_T）

定义
typedef sequence<CrossConnect_T> CrossConnectList_T;
说明

对象说明	交叉连接列表

5.1.9.14 资源信息（Resource_T）

定义
typedef globaldefs::NamingAttributes_T Resource_T;
说明

对象说明	资源信息

5.1.9.15 资源信息列表（ResourceList_T）

定义
typedef sequence<Resource_T> ResourceList_T;
说明

对象说明	资源信息列表，包含网元、拓朴连接、物理终端点、连接终端点、子网连接

5.1.9.16 路由信息（Route_T）

定义
typedef sequence<CrossConnect_T> Route_T;
说明

对象说明	路由信息

5.1.9.17 路由描述符信息（RouteDescriptor_T）

定义
struct RouteDescriptor_T { string id; string intended; string actualState; string administrativeState; string inUseBy; string exclusive;

定义
Route_T routeXCs; globaldefs::NVSList_T additionalInfo; };

说明	
对象说明	路由描述符信息
属性名	属性说明
id	标识符
intended	是否为期望路由，取值："y"、"n"
actualState	状态，取值："inactive"表示包含的交叉连接未激活；"active"：表示包含的交叉连接已激活；" partial"：表示包含的交叉连接部分激活
administrativeState	管理状态，取值"locked"表示路由不能被激活；"unlocked"表示路由可以被激活
inUseBy	是否已被其他连接使用，取值"y"、"n"
exclusive	是否独占，取值"y"、"n"
routeXCs	包含的交叉连接
additionalInfo	附加信息

5.1.9.18 子网连接路由列表（RouteList_T）

定义	
typedef sequence<RouteDescriptor_T> RouteList_T;	
说明	
对象说明	子网连接路由列表

5.1.9.19 SNC 创建数据（SNCCreateData_T）

定义

struct SNCCreateData_T {
string userLabel;
boolean forceUniqueness;
string owner;
globaldefs::ConnectionDirection_T direction;
StaticProtectionLevel_T staticProtectionLevel;
ProtectionEffort_T protectionEffort;
Reroute_T rerouteAllowed;
NetworkRouted_T networkRouted;
SNCType_T sncType;
transmissionParameters::LayerRate_T layerRate;
CrossConnectList_T ccInclusions;
ResourceList_T neTpInclusions;
boolean fullRoute;
ResourceList_T neTpSncExclusions;
globaldefs::NamingAttributesList_T aEnd;
globaldefs::NamingAttributesList_T zEnd;
globaldefs::NVSList_T additionalCreationInfo;
};

说明	
对象说明	SNC 创建数据
属性名	属性说明
userLabel	SNC 的用户标签
forceUniqueness	是否要求用户标签唯一
owner	SNC 的所有者
direction	SNC 的方向
staticProtectionLevel	SNC 的静态保护等级
protectionEffort	保护尽力程度
rerouteAllowed	重路由允许
networkRouted	网络层重路由允许
sncType	SNC 类型
layerRate	层速率
ccInclusions	必须包含的交叉连接
neTpInclusions	必须包含的资源
fullRoute	是否提供的完整路由
neTpSncExclusions	必须排除的资源
aEnd	SNC 的 A 点名称列表
zEnd	SNC 的 Z 点名称列表
additionalCreationInfo	附加信息

5.1.9.20 SNC 创建数据列表（SNCCreateDataList_T）

定义
typedef sequence<SNCCreateData_T> SNCCreateDataList_T;
说明
对象说明	SNC 创建数据列表

5.1.9.21 SNC 修改数据（SNCModifyData_T）

定义
struct SNCModifyData_T { string userLabel; boolean forceUniqueness; string owner; globaldefs::ConnectionDirection_T direction; string modifyType; boolean retainOldSNC; boolean modifyServers_allowed; StaticProtectionLevel_T staticProtectionLevel; ProtectionEffort_T protectionEffort; Reroute_T rerouteAllowed; NetworkRouted_T networkRouted; SNCType_T sncType; transmissionParameters::LayerRate_T layerRate; RouteList_T addedOrNewRoute;

定义
RouteList_T removedRoute; ResourceList_T neTpInclusions; boolean fullRoute; ResourceList_T neTpSncExclusions; globaldefs::NamingAttributesList_T aEnd; globaldefs::NamingAttributesList_T zEnd; globaldefs::NVSList_T additionalCreationInfo; };

说明	
对象说明	SNC 修改数据
属性名	属性说明
userLabel	SNC 的用户标签
forceUniqueness	是否要求用户标签唯一
owner	SNC 的所有者
direction	SNC 的方向
modifyType	修改类型，包括：“rerouting”“add_protection”“remove_protection”
retainOldSNC	是否请求 EMS 保留原来的 SNC 在挂起状态
modifyServers_allowed	是否允许修改服务层来实现保护限制
staticProtectionLevel	SNC 的静态保护等级
protectionEffort	保护尽力程度
rerouteAllowed	重路由允许
networkRouted	网络层重路由允许
sncType	SNC 类型
layerRate	层速率
addedOrNewRoute	增加或新建的路由
removedRoute	删除的路由
neTpInclusions	必须包含的资源
fullRoute	是否提供的完整路由
neTpSncExclusions	必须排除的资源
aEnd	SNC 的 A 点名称列表
zEnd	SNC 的 Z 点名称列表
additionalCreationInfo	附加信息

5.1.9.22 扩展网元交叉连接（EXCrossConnect_T）

定义
struct EXCrossConnect_T { globaldefs::NamingAttributes_T name; string userLabel; string nativeEMSName; string owner; boolean active; globaldefs::ConnectionDirection_T direction;

定义
SNCType_T ccType; globaldefs::NamingAttributesList_T aEndNameList; globaldefs::NamingAttributesList_T zEndNameList; globaldefs::NVSList_T additionalInfo; };

说明	
对象说明	扩展交叉连接信息
属性名	属性说明
name	网元交叉名称
userLabel	交叉连接用户标签
nativeEMSName	交叉连接本地名称
owner	交叉连接所有者
active	交叉连接的激活状态
direction	交叉连接的方向
ccType	交叉连接的类型
aEndNameList	交叉连接的 A 点名称列表
zEndNameList	交叉连接的 Z 点名称列表
additionalInfo	附加信息

5.1.9.23 交叉连接列表（EXCrossConnectList_T）

定义
typedef sequence<EXCrossConnect_T> EXCrossConnectList_T;

说明	
对象说明	交叉连接列表

5.1.9.24 路由信息（EXRoute_T）

定义
typedef sequence<EXCrossConnect_T> EXRoute_T;

说明	
对象说明	路由信息

5.1.9.25 CCIterator_I 接口

5.1.9.25.1 从迭代器中查询数据（next_n）

定义
boolean next_n(in unsigned long how_many, out CrossConnectList_T ccList) raises (globaldefs::ProcessingFailureException);

说明	
功能描述	从迭代器中查询 *n* 条记录。当 how_many≥迭代器剩余记录数时，系统在数据取完后应该自动销毁该迭代器对象
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	ccList：首次查询返回的交叉连接列表
操作异常	EXCPT_INTERNAL_ERROR

5.1.9.25.2 查询迭代器记录条数(getLength)

定义	
unsigned long getLength() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	查询迭代器中数据记录的总条数（注意这里指的是迭代器中数据记录的总条数，该值不随在迭代器生命周期内是不变的）
输入参数	无
输入/输出参数	无
输出参数	无
返回值	unsigned long：迭代器中包含的总的记录条数
操作异常	EXCPT_INTERNAL_ERROR

5.1.9.25.3 销毁迭代器对象(destroy)

定义	
void destroy() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	销毁迭代器对象
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR

5.1.9.26 SNCIterator_I接口

5.1.9.26.1 从迭代器中查询数据（next_n）

定义	
boolean next_n(in unsigned long how_many, out SubnetworkConnectionList_T sncList) raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	从迭代器中查询 *n* 条记录。当 how_many≥迭代器剩余记录数时，系统在数据取完后应该自动销毁该迭代器对象
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	sncList：首次查询返回的 SNC 列表
操作异常	EXCPT_INTERNAL_ERROR

5.1.9.26.2 查询迭代器记录条数（getLength）

定义
unsigned long getLength() raises (globaldefs::ProcessingFailureException);

说明	
功能描述	查询迭代器中数据记录的总条数（注意这里指的是迭代器中数据记录的总条数，该值不随在迭代器生命周期内是不变的）
输入参数	无
输入/输出参数	无
输出参数	无
返回值	unsigned long：迭代器中包含的总的记录条数
操作异常	EXCPT_INTERNAL_ERROR

5.1.9.26.3 销毁迭代器对象（destroy）

定义	
void destroy() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	销毁迭代器对象
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR

5.1.9.27 EXCCIterator_I 接口

5.1.9.27.1 从迭代器中查询数据（next_n）

定义	
boolean next_n(in unsigned long how_many, out EXCrossConnectList_T exccList) raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	从迭代器中查询 *n* 条记录。当 how_many≥迭代器剩余记录数时，系统在数据取完后应该自动销毁该迭代器对象
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	exccList：首次查询返回的交叉连接列表
操作异常	EXCPT_INTERNAL_ERROR

5.1.9.27.2 查询迭代器记录条数(getLength)

定义	
unsigned long getLength() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	查询迭代器中数据记录的总条数（注意这里指的是迭代器中数据记录的总条数，该值不随在迭代器生命周期内是不变的）
输入参数	无
输入/输出参数	无
输出参数	无
返回值	unsigned long：迭代器中包含的总的记录条数
操作异常	EXCPT_INTERNAL_ERROR

5.1.9.27.3 销毁迭代器对象(destroy)

定义	
void destroy() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	销毁迭代器对象
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR

5.1.10 流域模块(module flowDomain)

说明：对于端到端业务的管理可以采用 SNC 或 FDFr 中任意一种数据格式表达，不根据业务类型进行区分。

5.1.10.1 连接状态（ConnectivityState_T）

定义	

enum ConnectivityState_T
{
CS_UNKNOW,
CS_FULLY_CONNECTED,
CS_NOT_FULLY_CONNECTED
};	

说明	
对象说明	子网的拓扑类型
类型取值	取值说明
CS_UNKNOW	未知
CS_FULLY_CONNECTED	全连接（所有 FD 的边缘点均是互相连接的）
CS_NOT_FULLY_CONNECTED	部分连接（不是所有的 FD 的边缘点都是互相连接的，即至少有一个 FD 的边缘点和另一个 FD 的边缘点无连接）

5.1.10.2 流域类型（FDType_T）

定义	

enum FDType_T
{
FDT_SINGLETON,
FDT_NETWORK
};	

说明	
对象说明	流域类型
类型取值	取值说明
FDT_SINGLETON	FD 下只含有一个 MFD
FDT_NETWORK	FD 下不止含有一个 MFD

5.1.10.3 流域(FlowDomain_T)

定义

struct FlowDomain_T {
globaldefs::NamingAttributes_T name;
string userLabel;
string nativeEMSName;
string owner;
transmissionParameters::LayeredParameterList_T transmissionParams;
string networkAccessDomain;
ConnectivityState_T fDConnectivityState;
FDType_T fdType;
globaldefs::NVSList_T additionalInfo;
};

说明	
对象说明	流域信息
属性名	属性说明
name	流域名称
userLabel	流域的用户标签
nativeEMSName	流域的本地名称
owner	流域的所有者
transmissionParams	层速率及流域参数列表
networkAccessDomain	包含该流域的网络接入域
fDConnectivityState	流域下的分块域间的服务层连接状态
fdType	流域类型
additionalInfo	附加信息

5.1.10.4 FDIterator_I 接口

5.1.10.4.1 从迭代器中查询数据(next_n)

定义
boolean next_n(in unsigned long how_many, out FDList_T fdList) raises (globaldefs::ProcessingFailureException);

说明	
功能描述	从迭代器中查询 *n* 条记录。当 how_many≥迭代器剩余记录数时，系统在数据取完后应该自动销毁该迭代器对象
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	fdList：首次查询返回的流域列表
操作异常	EXCPT_INTERNAL_ERROR

5.1.10.4.2 查询迭代器记录条数(getLength)

定义
unsigned long getLength() raises (globaldefs::ProcessingFailureException);

说明	
功能描述	查询迭代器中数据记录的总条数（注意这里指的是迭代器中数据记录的总条数，该值在迭代器生命周期内是不变的）
输入参数	无
输入/输出参数	无
输出参数	无
返回值	unsigned long：迭代器中包含的总的记录条数
操作异常	EXCPT_INTERNAL_ERROR

5.1.10.4.3 销毁迭代器对象(destory)

定义	
void destroy() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	销毁迭代器对象
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR

5.1.10.5 FlowDomainMgr_I 接口

5.1.10.5.1 查询 EMS 下所有的流域（getAllFlowDomains）

定义	
void getAllFlowDomains(in unsigned long how_many, out FDList_T flowDomains, out FDIterator_I fdIt) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询 EMS 下所有的流域
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	flowDomains：流域列表。 fdIt：迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.10.5.2 查询 EMS 下所有的 FDFrs(getAllFDFrs)

定义
void getAllFDFrs(in globaldefs::NamingAttributes_T fdName, in unsigned long how_many, in transmissionParameters::LayerRateList_T connectivityRateList, out flowDomainFragment::FDFrList_T fdfrList,

定义	
out flowDomainFragment::FDFrIterator_I fdfrIt) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询 EMS 下所有的 FDFrs,包括端到端以太网业务和端到端 L3VPN 业务
输入参数	fdName：流域名称。 how_many：首次迭代查询返回的数据数目。 connectivityRateList：连接层速率列表
输入/输出参数	无
输出参数	fdfrList：FDFr 列表。 fdfrIt：迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.10.5.3 查询指定的 FDFr(getFDFr)

定义	
void getFDFr(in globaldefs::NamingAttributes_T fdfrName, out flowDomainFragment::FlowDomainFragment fdfr) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询指定的 FDFr
输入参数	fdfrName：FDFr 名称
输入/输出参数	无
输出参数	fdfr：FDFr 信息
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.1.10.5.4 创建并激活 FdFr(createAndActivateFDFr)

定义
void createAndActivateFDFr(in flowDomainFragment::FDFrCreateDaTa_T createData, in ConnectivityRequirement_T connectivityRequirement, in globaldefs::NamingAttributesList_T aEnd, in globaldefs::NamingAttributesList_T zEnd, inout globaldefs::NamingAttributesList_T internalTPs, inout flowDomainFragment::MatrixFlowDomainFragmentList_T mfdfrs, inout subnetworkConnection::TPDataList_T tpsToModify, out flowDomainFragment::FlowDomainFragment_T theFDFr,

定义	
out globaldefs::NamingAttributesList_T notConnectableCPTPList, out globaldefs::NamingAttributesList_T parameterProblemsTPList, out string errorReason) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	创建并激活 FDFr
输入参数	createData：创建 FDFr 数据信息 connectivityRequirement：连接请求参数。若 EMS 本身具备识别连接的能力，此参数用于区分 FP 是否可以连接到其他的 FP；若 EMS 本身不具备识别连接的能力，此参数无效。 aEnd：待创建的 FDFr 的 A 端点信息。 zEnd：待创建的 FDFr 的 Z 端点信息
输入/输出参数	internalTPs：FDFr 的路由端点信息，此参数可以为空。 Mfdfrs：组成 FDFr 的 MFDFr。 tpsToModify：TP 点以及相应的参数信息
输出参数	theFDFr：新建的 FDFr 信息。 notConnectableCPTPList：不可连接的端点信息。 parameterProblemsTPList：连接终端点及流点列表。其只有尽力传送参数不可设置。 errorReason：失败原因
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.1.10.5.5 删除并去激活 FdFr(deactivateAndDeleteFDFr)

定义	
void deactivateAndDeleteFDFr(in globaldefs::NamingAttributes_T fdfrName, inout subnetworkConnection::TPDataList_T tpsToModify, out flowDomainFragment::FlowDomainFragment_T theFDFr, out string errorReason) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	删除并去激活指定的 FDFr
输入参数	fdfrName：指定的待删除和去激活的 FDFr 名称。 tpsToModify：TP 点以及相应的参数信息
输入/输出参数	无
输出参数	theFDFr：被删除和去激活的 FDFr 信息。 errorReason：失败原因
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_ACCESS_DENIED EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.1.10.5.6 修改 FdFr(modifyFDFr)

定义
void modifyFDFr(in globaldefs::NamingAttributes_T fdfrName, in flowDomainFragment::FDFrModifyData_T fdfrModifyData, in ConnectivityRequirement_T connectivityRequirement, inout subnetworkConnection::TPDataList_T tpsToModify, out globaldefs::NamingAttributes_T failedTPList, out globaldefs::NamingAttributes_T parameterProblemsTPList, out flowDomainFragment::FlowDomainFragment_T newFDFr, out string errorReason) raises(globaldefs::ProcessingFailureException);

说明	
功能描述	修改指定的 FDFr
输入参数	fdfrName：指定的待修改的 FDFr 名称。 fdfrModifyData：描述 FDFr 需要修改的数据信息。 connectivityRequirement：连接请求参数。若 EMS 本身具备识别连接的能力，此参数用于区分 FP 是否可以连接到其他的 FP；若 EMS 本身不具备识别连接的能力，此参数无效
输入/输出参数	tpsToModify：TP 点以及相应的参数信息
输出参数	failedTPList：失败的 TP 列表。 parameterProblemsTPList：连接终端点及流点列表。其只有尽力传送参数不可设置。 newFDFr：修改后的 FDFr 信息。 errorReason：失败原因
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_ACCESS_DENIED EXCPT_UNABLE_TO_COMPLY EXCPT_NOT_IN_VALID_INPUT

5.1.10.5.7 查询 FDFr 的路由信息（getFDFrRoute）

定义
void getFDFrRoute(in globaldefs::NamingAttributes_T fdfrName, out flowDomainFragment::FDFrRoute_T route) raises(globaldefs::ProcessingFailureException);

说明	
功能描述	查询 FDFr 的路由信息
输入参数	fdfrName：指定的 FDFr 名称
输入/输出参数	无
输出参数	route：返回指定的 FDFr 的路由信息
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.1.10.5.8 创建 MFDFr（createEXMFDFr）

定义	
void createEXMFDFr(in globaldefs::NamingAttributes_T meName, in EXMatrixFlowDomainFragment_T exMFdFrToCreate, in EXFdFrCreateAdditionalData_T fdFrCreateAdditionInfo) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	创建 MFDFr
输入参数	meName：指定需要创建流域片段的网元名称。 exMFdFrToCreate：创建 MFdFr 的数据信息。 fdFrCreateAdditionInfo：创建流域片段的附加信息
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

5.1.10.5.9 激活 MFDFr（activateEXMFDFr）

定义	
void activateEXMFDFr(in globaldefs::NamingAttributes_T exMFdFrName) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	激活指定的 MFDFr 列表
输入参数	exMFdFrName：指定的待激活的 MFdFr 名称
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

5.1.10.5.10 去激活 MFdFr（deActivateEXMFDFr）

定义	
void deActivateEXMFDFr(in globaldefs::NamingAttributes_T exMFdFrName) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	去激活指定的 MFDFr
输入参数	exMFdFrName：指定的待去激活的 MFdFr 名称
输入/输出参数	无

说明	
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

5.1.10.5.11 删除 MFDFr(deleteEXMFDFrs)

定义	
void deleteEXMFDFrs(in globaldefs::NamingAttributes_T exMFdFrNames) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	删除指定的 MFDFr
输入参数	exMFdFrNames：指定的待删除的 MFdFr 名称列表
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

5.1.10.5.12 激活 FDFr(activateFDFr)

定义	
void activateFDFr(in globaldefs::NamingAttributes_T fdfrName) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	激活指定的 FDFr
输入参数	fdfrName：指定待激活的 FdFr 名称
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

5.1.10.5.13 去激活 FDFr（deActivateFDFr）

定义	
void deActivateFDFr(in globaldefs::NamingAttributes_T fdfrName) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	去激活指定的 FDFr 列表
输入参数	fdfrNames：指定的待去激活的 FdFr 名称列表
输入/输出参数	无

说明	
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

5.1.10.5.14 删除 FDFr（deleteFDFr）

定义	
void deleteFDFr(in globaldefs::NamingAttributes_T fdfrName) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	删除指定的 FDFr
输入参数	fdfrName：指定的待删除的 FdFr 名称
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

5.1.10.5.15 创建 FDFr（createFDFr）

定义	
void createFDFr(in string createType, in flowDomainFragment::FlowDomainFragment_T fdFrToCreate, in EXFDFrCreateAdditionalData_T fdFrCreateAdditionInfo,) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	创建 FDFr
输入参数	createType：创建 FDFr 的类型。 fdFrToCreate：创建 FDFr 需要的信息。 fdFrCreateAdditionInfo：创建 FDFr 的附加信息
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

5.1.10.5.16 修改 FDFr(modifyEXFDFr)

定义
void modifyEXFDFr(in string modifyType, in flowDomainFragment::FlowDomainFragment_T fdFrToModify,

定义	
in EXFdFrCreateAdditionalData_T fdFrModifyAdditionInfo, out flowDomainFragment::FlowDomainFragment_T fdFrSuccess) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	修改 FDFr
输入参数	modifyType：待修改的 FDFr 的类型。 fdFrToModify：待修改的 FDFr 信息。 fdFrModifyAdditionInfo：待修改的 FDFr 的附加信息
输入/输出参数	无
输出参数	fdFrSuccess：修改成功后的 FDFr 信息
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

5.1.10.5.17 查询 FDFr 的路由（getEXFDFrRoute）

定义	
void getEXFDFrRoute(in globaldefs::NamingAttributes_T fdfrName, out EXMatrixFlowDomainFragmentList_T route, raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询 FDFr 的路由信息
输入参数	fdfrName：待查的 FDFr 名称
输入/输出参数	无
输出参数	route：FDFr 的路由信息
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

5.1.10.5.18 查询网元下所有的 MFDFrs（getAllEXMFDFrs）

定义	
void getAllEXMFDFrs(in globaldefs::NamingAttributes_T meName, in transmissionParameters::LayerRateList_T layerList, in unsigned long how_many, out EXMatrixFlowDomainFragmentList_T exMFdFrList, out EXMFDFrIterator_T exMFdFrIt) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询指定网元下所有的 MFDFr
输入参数	meName：指定的网元名称。 layerList：速率层次列表。允许为空。 how_many：首次迭代查询返回的数据数目

说明	
输入/输出参数	无
输出参数	exMFdFrList：MFDFr 列表信息。 exMFdFrIt：迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

5.1.10.5.19 网络业务路由调整（adjustFDFrRoute）（可选）

定义
void adjustmentFDFrRoute(in globaldefs::NamingAttributes_T fdfrNameToAdjust, in EXFdFrCreateAdditionalData_T fdfrAdjustParam, in EXFdFrCreateAdditionalData_T fdfrAdjustOrder, out flowDomainFragment::FlowDomainFragment_T fdfrSuccess) raises(globaldefs::ProcessingFailureException);

说明	
功能描述	调整指定业务的路由信息
输入参数	fdfrNameToAdjust：指定的业务名称。 fdfrAdjustParam：增加的路由信息以及删除的路由信息。 fdfrAdjustOrder：完整的路由信息，便于确定增加的路由顺序
输入/输出参数	无
输出参数	fdfrSuccess：调整后的业务信息
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

5.1.11 FDFr 模块

5.1.11.1 EXMFDFr（EXMatrixFlowDomainFragment_T）

定义

struct EXMatrixFlowDomainFragment_T {
globaldefs::NamingAttributes_T exMFdFrName,
string userLabel,
string nativeEMSName,
string owner,
globaldefs::ConnectionDirection_T direction;
transmissionParameters::LayeredParameters_T transmissionParams;
subnetworkConnection::TPDataList_T aEnd;
subnetworkConnection::TPDataList_T zEnd;
boolean flexible;
boolean active;
FDFrType_T mfdfrType;
globaldefs::NVSList_T additionalInfo;
};

说明	
对象说明	MFDFr 信息
属性名	属性说明
exMFdFrName	MFDFr 名称
userLabel	MFDFr 的用户标签
nativeEMSName	MFDFr 的本地名称
owner	MFDFr 的所有者
direction	MFDFr 的方向
transmissionParams	层速率及参数列表
aEnd	MFDFr 的 A 端点
zEnd	MFDFr 的 Z 端点
flexible	标识 MFDFr 是否是可调节的
active	激活标识
mfdfrType	MFDFr 类型
additionalInfo	附加信息

5.1.11.2 FDFr（FlowDomainFragment_T）

定义

struct FlowDomainFragment_T {
globaldefs::NamingAttributes_T name;
string userLabel;
string nativeEMSName;
string owner;
globaldefs::ConnectionDirection_T direction;
transmissionParameters::LayeredParameters_T transmissionParams;
subnetworkConnection::TPDataList_T aEnd;
subnetworkConnection::TPDataList_T zEnd;
string networkAccessDomain;
boolean flexible;
performance::AdministrativeState_T administrativeState;
subnetworkConnection::SNCState_T fdfrState;
FDFrType_T fdfrType;
globaldefs::NVSList_T additionalInfo;
};

说明	
对象说明	FDFr 信息
属性名	属性说明
name	FDFr 名称
userLabel	FDFr 的用户标签
nativeEMSName	FDFr 的本地名称
owner	FDFr 的所有者
direction	FDFr 的方向
transmissionParams	层速率及参数列表
aEnd	FDFr 的 A 端点
zEnd	FDFr 的 Z 端点

说明	
networkAccessDomain	网络接入域
flexible	标识 FDFr 是否是可调节的
administrativeState	FDFr 锁定标识
fdfrState	FDFr 激活标识
fdfrType	FDFr 类型
additionalInfo	附加信息

5.1.11.3 FDFr 类型 （FDFrType_T）

定义

enum FDFrType_T
{
FDFRT_POINT_TO_POINT,
FDFRT_POINT_TO_MULTIPOINT,
FDFRT_MULTIPOINT
};

说明	
对象说明	FDFr 类型
类型取值	取值说明
FDFRT_POINT_TO_POINT	点到点
FDFRT_POINT_TO_MULTIPOINT	点到多点
FDFRT_MULTIPOINT	多点到多点

5.1.11.4 FDFr 创建数据（FDFrCreateData_T）

定义

struct FDFrCreateData_T{
globaldefs::NamingAttributes_T name;
string userLabel;
boolean forceUniqueness;
string owner;
string networkAccessDomain;
globaldefs::ConnectionDirection_T direction;
performance::AdministrativeState_T administrativeState;
transmissionParameters::LayeredParameters_T transmissionParams;
boolean fullRoute;
FDFrType_T fdfrType;
globaldefs::NVSList_T additionalCreationInfo;
};

说明	
对象说明	FDFr 创建数据
类型取值	取值说明
name	FDFr 名称
userLabel	FDFr 用户标签，可以为空
forceUniqueness	标识符，用于描述 FDFr 的用户标签是否唯一

说明	
owner	FDFr 所有者
networkAccessDomain	网络接入域
direction	FDFr 方向。注：以太网层的 FDFr 始终为双向
administrativeState	FDFr 锁定标识
transmissionParams	FDFr 的速率层次以及相应的参数信息
fullRoute	标识符，描述是否全路由标识
fdfrType	FDFr 类型
additionalCreationInfo	附加信息

5.1.11.5 FDFr 修改数据信息（FDFrModifyData_T）

定义

struct FDFrModifyData_T {
string userLabel;
boolean forceUniqueness;
string owner;
string networkAccessDomain;
performance::AdministrativeState_T administrativeState;
transmissionParameters::LayeredParameters_T transmissionParams;
globaldefs::NamingAttributesList_T tpNamesToRemove;
globaldefs::NamingAttributesList_T aEndTPNames;
globaldefs::NamingAttributesList_T zEndTPNames;
globaldefs::NamingAttributesList_T internalTPNames;
globaldefs::NVSList_T additionalModificationInfo;
};

说明	
对象说明	FDFr 创建数据
类型取值	取值说明
userLabel	FDFr 用户标签，可以为空
forceUniqueness	标识符，用于描述 FDFr 的用户标签是否唯一
owner	FDFr 所有者
networkAccessDomain	网络接入域
administrativeState	FDFr 锁定标识
transmissionParams	FDFr 的速率层次以及相应的参数信息
tpNamesToRemove	FDFr 中需要删除的 TP 点信息
aEndTPNames	源端点名称列表
zEndTPNames	宿端点名称列表
internalTPNames	内部端点信息，用于描述 FDFr 的路由端点信息
additionalModificationInfo	附加信息

5.1.11.6 MFDFr(MatrixFlowDomainFragment_T)

定义
struct MatrixFlowDomainFragment_T { globaldefs::ConnectionDirection_T direction; transmissionParameters::LayeredParameters_T transmissionParams;

定义
subnetworkConnection::TPDataList_T aEnd; subnetworkConnection::TPDataList_T zEnd; boolean flexible; Boolean active; FDFrType_T mfdfrType; globaldefs::NVSList_T additionalInfo; };

说明	
对象说明	FDFr 信息
属性名	属性说明
direction	MFDFr 的方向
transmissionParams	层速率及参数列表
aEnd	MFDFr 的 A 端点
zEnd	MFDFr 的 Z 端点
flexible	标识 MFDFr 是否是可调节的
active	FDFr 激活标识
mfdfrType	MFDFr 类型
additionalInfo	附加信息

5.1.11.7 FDFr 路由信息（FDFrRoute_T）

定义	
typedef sequence<MatrixFlowDomainFragment_T> FDFrRoute_T;	
说明	
对象说明	MFDFr 列表

5.1.11.8 EXMFDFr 列表（EXMatrixFlowDomainFragmentList_T）

定义	
typedef sequence<EXMatrixFlowDomainFragment_T> EXMatrixFlowDomainFragmentList_T;	
说明	
对象说明	ExMFDFr 列表

5.1.11.9 EXFDFr 路由列表（EXFDFrRoute_T）

定义	
typedef sequence<EXMatrixFlowDomainFragment_T> EXFDFrRoute_T;	
说明	
对象说明	MFDFr 列表

5.1.11.10 FDFr 列表（FDFrList_T）

定义	
typedef sequence<FlowDomainFragment_T> FDFrList_T;	
说明	
对象说明	FDFr 列表

5.1.11.11 创建 FDFr 所需的附加参数信息（EXFDFrCreateAdditionalData_T）

定义

struct EXFDFrCreateAdditionalData_T {
boolean fullRoute;
boolean forceUniqueness;
subnetworkConnection::Reroute_T rerouteAllowed;
subnetworkConnection::NetworkRouted_T networkRouted:
transmissionParameters::LayerRate_T layerRate;
subnetworkConnection::ResourceList_T inclusionResource;
subnetworkConnection::ResourceList_T exclusionResource;
EXMatrixFlowDomainFragmentList_T ccInclusions;
EXMatrixFlowDomainFragmentList_T backupCCInclusions;
globaldefs::NVSList_T additionalCreationInfo;
};

说明	
对象说明	创建 FDFr 所需的附加参数信息
属性名	属性说明
fullRoute	完整路由信息标识。当取值为 true 时，表示创建时需指定全路由
forceUniqueness	标识 userlabel 是否唯一。若不唯一则不创建
rerouteAllowed	重路由标识
networkRouted	网络路由标识
layerRate	层速率。指定创建哪个层次的流域
inclusionResource	必须经过的资源
exclusionResource	必须绕过的资源
ccInclusions	若 fullRoute 填为 true，则此处必须填写完整的主用 MFDFr 信息
backupCCInclusions	若 fullRoute 填为 true，则此处必须填写完整的备用 MFDFr 信息
additionalCreationInfo	附加参数信息

5.1.11.12 EXMFDFrIterator_I 接口

5.1.11.12.1 从迭代器中查询数据（next_n）

定义
boolean next_n(in unsigned long how_many, out EXMatrixFlowDomainFragmentList_T mfdfrList) raises (globaldefs::ProcessingFailureException);

说明	
功能描述	从迭代器中查询 *n* 条记录。当 how_many≥迭代器剩余记录数时，系统在数据取完后应该自动销毁该迭代器对象
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	mfdfrList：首次查询返回的分块流域列表
操作异常	EXCPT_INTERNAL_ERROR

5.1.11.12.2 查询迭代器记录条数（getLength）

定义	
unsigned long getLength() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	查询迭代器中数据记录的总条数（注意这里指的是迭代器中数据记录的总条数，该值在迭代器生命周期内是不变的）
输入参数	无
输入/输出参数	无
输出参数	无
返回值	unsigned long：迭代器中包含的总的记录条数
操作异常	EXCPT_INTERNAL_ERROR

5.1.11.12.3 销毁迭代器对象（destory）

定义	
void destroy() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	销毁迭代器对象
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR

5.1.11.13 FDFrIterator_I 接口

5.1.11.13.1 从迭代器中查询数据（next_n）

定义	
boolean next_n(in unsigned long how_many, out FDFrList_T fdfrList) raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	从迭代器中查询 *n* 条记录。当 how_many≥迭代器剩余记录数时，系统在数据取完后应该自动销毁该迭代器对象
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	fdList：首次查询返回的流域列表
操作异常	EXCPT_INTERNAL_ERROR

5.1.11.13.2 销毁迭代器记录条数（getLength）

定义	
unsigned long getLength() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	查询迭代器中数据记录的总条数（注意这里指的是迭代器中数据记录的总条数，该值在迭代器生命周期内是不变的）
输入参数	无

说明	
输入/输出参数	无
输出参数	无
返回值	unsigned long：迭代器中包含的总的记录条数
操作异常	EXCPT_INTERNAL_ERROR

5.1.11.13.3 销毁迭代器对象（destroy）

定义	
void destroy() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	销毁迭代器对象
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR

5.1.12 业务割接计划模块（可选）

5.1.12.1 业务割接计划（BusinessCutOverScheme_T）

定义	

struct BusinessCutOverScheme_T {
globaldefs::NamingAttributes_T name;
string userLabel;
string nativeEMSName;
string owner;
string mode;
string policy;
globaldefs::Time_T cutOverTime;
globaldefs::NVSList_T additionalInfo;
};	

说明	
对象说明	业务割接计划
属性名	属性说明
name	业务割接计划名称
userLabel	业务割接计划的用户标签
nativeEMSName	业务割接计划的本地名称
owner	业务割接计划的所有者
mode	业务割接模式
policy	业务割接策略
cutOverTime	业务割接时间
additionalInfo	附加信息

5.1.12.2 业务割接计划列表（BusinessCutOverSchemeList_T）

定义	
typedef sequence<BusinessCutOverPlan_T> BusinessCutOverSchemeList_T;	
说明	
对象说明	业务割接计划列表

5.1.12.3 业务割接计划创建数据（BusinessCutOverSchemeCreateData_T）

定义

struct BusinessCutOverSchemeCreateData_T {
string nativeEMSName;
string userLabel;
string mode;
string policy;
globaldefs::Time_T cutOverTime;
globaldefs::NVSList_T additionalInfo;
};

说明	
对象说明	业务割接计划
属性名	属性说明
nativeEMSName	业务割接计划的本地名称
userLabel	业务割接计划的用户标签
mode	业务割接模式。描述人工割接还是自动割接
policy	业务割接策略。描述尽量割接还是回滚割接
cutOverTime	业务割接时间
additionalInfo	附加信息

5.1.12.4 业务割接 BusinessCutOverSchemeIterator_I 接口

5.1.12.4.1 从迭代器中查询数据（next_n）

定义	
boolean next_n(in unsigned long how_many, out BusinessCutOverPlanList_T bcoPlanList) raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	从迭代器中查询 *n* 条记录。当 how_many≥迭代器剩余记录数时，系统在数据取完后应该自动销毁该迭代器对象
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	fdList：首次查询返回的流域列表
操作异常	EXCPT_INTERNAL_ERROR

5.1.12.4.2 销毁迭代器记录条数（getLength）

定义	
unsigned long getLength() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	查询迭代器中数据记录的总条数（注意这里指的是迭代器中数据记录的总条数，该值在迭代器生命周期内是不变的）
输入参数	无
输入/输出参数	无
输出参数	无

说明	
返回值	unsigned long：迭代器中包含的总的记录条数
操作异常	EXCPT_INTERNAL_ERROR

5.1.12.4.3 销毁迭代器对象（destroy）

定义	
void destroy() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	销毁迭代器对象
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR

5.1.12.5 业务割接计划接口

5.1.12.5.1 创建业务割接计划（createBusinessCutOverScheme）

定义	
void createBusinessCutOverScheme(in BusinessCutOverSchemeCreateData_T bcoSchemeCreateData, out BusinessCutOverScheme_T bcoResult, out string errorReason) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	创建业务割接计划
输入参数	bcoSchemeCreateData：业务割接计划创建参数信息
输入/输出参数	无
输出参数	bcoResult：创建后的业务割接计划信息。 errorReason：创建失败的原因
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.1.12.5.2 查询 EMS 下所有业务割接计划（getAllBusinessCutOverSchemes）

定义	
void getAllBusinessCutOverSchemes(in globaldefs::NamingAttribute_T emsName, in unsigned long how_many, out BusinessCutOverSchemeList_T bcoSchemeList, out BusinessCutOverSchemeIterator_I bcoSchemeIt) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询 EMS 下所有业务割接计划
输入参数	emsName：EMS 名称。 how_many：首次迭代查询返回的数据数目

说明	
输入/输出参数	无
输出参数	bcoSchemeList：业务割接计划列表。 bcoSchemeIt：迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.1.12.5.3 删除指定的业务割接计划（deleteBusinessCutOverScheme）

定义	
void deleteBusinessCutOverScheme(in globaldefs::NamingAttribute_T bcoSchemeName) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	删除指定的业务割接计划
输入参数	bcoSchemeName：业务割接名称
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.1.12.5.4 执行业务割接计划（executeBusinessCutOverScheme）

定义	
void executeBusinessCutOverScheme(in globaldefs::NamingAttributes_T bcoSchemeName) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	执行业务割接计划
输入参数	bcoSchemeName：业务割接计划名称
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.1.12.5.5 查询业务割接结果（getResultofBusinessCutOverScheme）

定义
void getResultofBusinessCutOverScheme(in globaldefs::NamingAttributes_T bcoSchemeName,

定义
out globaldefs::Time_T executeTime, out NamingAttributesMultipleList_T failedNamePairList, out NamingAttributesMultipleList_T successNamePairList, out globaldefs::NVSList_T additionalInfo) raises(globaldefs::ProcessingFailureException);

说明	
功能描述	查询业务割接结果
输入参数	bcoSchemeName：业务割接计划名称
输入/输出参数	无
输出参数	executeTime：业务割接时间。 failedNamePairList：割接失败的业务对列表。 successNamePairList：割接成功的业务对列表。 additionalInfo：附加信息
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.1.13 业务割接组模块

5.1.13.1 业务割接组（BusinessCutOverGroup_T）

定义

struct BusinessCutOverGroup_T {
globaldefs::NamingAttributes_T name;
string userLabel;
string nativeEMSName;
globaldefs::NamingAttributes_T businessCutOverSchemeName;
NamingAttributesMultipleList_T businessCutOverSncList;
globaldefs::NVSList_T additionalInfo;
};

说明	
对象说明	业务割接计划
属性名	属性说明
name	业务割接组名称
userLabel	业务割接组的用户标签
nativeEMSName	业务割接组的本地名称
businessCutOverSchemeName	业务割接组所属的割接计划标识
businessCutOverSncList	割接业务标识列表
additionalInfo	附加信息

5.1.13.2 业务割接组列表（BusinessCutOverGroupList_T）

定义	
typedef sequence<BusinessCutOverGroup_T> BusinessCutOverGroupList_T;	
说明	
对象说明	业务割接组列表

5.1.13.3 业务割接组创建数据（BusinessCutOverGroupCreateData_T）

定义

struct BusinessCutOverGroup_T {
string userLabel;
string nativeEMSName;
globaldefs::NamingAttributes_T businessCutOverSchemeName;
NamingAttributesMultipleList_T businessCutOverSncList;
globaldefs::NVSList_T additionalInfo;
};

说明	
对象说明	业务割接计划
属性名	属性说明
userLabel	业务割接组的用户标签
nativeEMSName	业务割接组的本地名称
businessCutOverSchemeName	业务割接组所属的割接计划标识
businessCutOverSncList	割接业务标识列表
additionalInfo	附加信息

5.1.13.4 业务割接组接口

5.1.13.4.1 创建业务割接组（createBusinessCutOverGroup）

定义	
void createBusinessCutOverGroup(in BusinessCutOverGroupCreateData_T bcoGroupCreateData, out BusinessCutOverGroup_T bcoGroupResult) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	创建业务割接组
输入参数	bcoGroupCreateData：待创建的业务割接组信息
输入/输出参数	无
输出参数	bcoGroupResult：创建的业务割接组信息
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.1.13.4.2 查询 EMS 下所有的业务割接组（getAllBusinessCutOverGroups）

定义
void getAllBusinessCutOverGroups (in globaldefs::NamingAttributes_T bcoSchemeName, out BusinessCutOverGroupList_T bcoGroupList) raises(globaldefs::ProcessingFailureException);

说明	
功能描述	取 EMS 下所有的业务割接组
输入参数	bcoSchemeName：业务割接计划名称
输入/输出参数	无
输出参数	bcoGroupList：业务割接组列表
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.1.13.4.3 修改业务割接组（modifyBusinessCutOverGroup）

定义	
void modifyBusinessCutOverGroup(in BusinessCutOverGroup_T bcoGroup, out BusinessCutOverGroup_T newBcoGroup) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	修改业务割接组
输入参数	bcoGroup：待修改的业务割接组信息
输入/输出参数	无
输出参数	newBcoGroup：修改后的业务割接组信息
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.1.13.4.4 删除业务割接组（deleteBusinessCutOverGroup）

定义	
void deleteBusinessCutOverGroup(in globaldefs::NamingAttributes_T bcoGroupName) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	删除业务割接组
输入参数	bcoGroupName：业务割接组名称
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.1.14 流量控制模板模块（module trafficConditioningProfile）

5.1.14.1 流量控制模板 TCProfile_T

定义
struct TCProfile_T { globaldefs ::NamingAtrributes_T name , string userLabel, string nativeEMSName, string owner, boolean defaultProfile, transmissionParameters ::LayeredParameterList_T transmissionParams, globaldefs ::NVSList_T additionalInfo, };

说明	
对象说明	流量控制模板
类型取值	取值说明
name	流量控制模板名称
userLabel	流量控制模板的用户标签
nativeEMSName	流量控制模板的本地名称
owner	流量控制模板的所有者
defaultProfile	默认模板标识。当该模板为默认模板时，表示该模板是不可删除的。一般 EMS 含多个 TC 模板来控制不同的 TP 点
transmissionParams	流量控制参数列表
additionalInfo	附加信息

5.1.14.2 流量控制模板列表 TCProfileList_T

定义
typedef sequence<TCProfile_T> TCProfileList_T;

说明	
对象说明	流量控制模板列表

5.1.14.3 TC 模板创建数据 TCProfileCreateData_T

定义
struct TCProfileCreateData_T { string userLabel, Boolean forceUniqueness, string owner, transmissionParameters ::LayeredParameterList_T transmissionParams, globaldefs ::NVSList_T additionalCreationInfo, };

说明	
对象说明	TC 模板创建数据
类型取值	取值说明
userLabel	TC 模板数据用户标签
forceUniqueness	标识 TC 模板用户标签是否唯一，若用户标签已经存在则此操作返回失败
owner	TC 模板的所有者
transmissionParams	流量控制参数列表
additionalCreationInfo	附加信息

5.1.14.4 模板分配任务 TCProfileAssignTask_T

定义
struct TCProfileAssignTask_T { globaldefs::NamingAttributes_T tcProfileName; globaldefs::NamingAttributesList_T entityNames; transmissionParameters::LayerRate_T layer; string dataType; globaldefs::NVSList_T additionalInfo; };

说明	
对象说明	分配 TC 模板数据
类型取值	取值说明
tcProfileName	TC 模板名称
entityNames	资源对象标识符，用于描述 TC 模板需要分配到的对象标识
layer	层速率，描述模板应用的层次
dataType	报文数据类型，用于区分用户报文和 OAM 报文
additionalCreationInfo	附加信息

5.1.14.5 TCProfileIterator_I 接口

5.1.14.5.1 从迭代器中查询数据（next_n）

定义
boolean next_n(in unsigned long how_many, out TCProfileList_T tcProfileList) raises (globaldefs::ProcessingFailureException);

说明	
功能描述	从迭代器中查询 *n* 条记录。当 how_many≥迭代器剩余记录数时，系统在数据取完后应该自动销毁该迭代器对象。
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	subnetworkList：首次查询返回的流量控制模板列表
操作异常	EXCPT_INTERNAL_ERROR

5.1.14.5.2 查询迭代器记录条数（getLength）

定义
unsigned long getLength() raises (globaldefs::ProcessingFailureException);

说明	
功能描述	查询迭代器中数据记录的总条数（注意这里指的是迭代器中数据记录的总条数，该值在迭代器生命周期内是不变的）
输入参数	无
输入/输出参数	无
输出参数	无
返回值	unsigned long：迭代器中包含的总的记录条数
操作异常	EXCPT_INTERNAL_ERROR

5.1.14.5.3 销毁迭代器对象（destory）

定义	
void destroy() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	销毁迭代器对象
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR

5.1.14.6 TCProfileMgr_I 接口

5.1.14.6.1 查询所有的 TC 模板（getAllTCProfiles）

定义	
void getAllTCProfiles(in unsigned long how_many, out TCProfileList_T tcProfileList, out TCProfileIterator_I tcProfileIt) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询 TC 模板管理器下的所有的 TC 模板信息
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	tcProfileList：TC 模板列表。 tcProfileIt：迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_NOT_IMPLEMENTED EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.14.6.2 查询指定的 TC 模板（getTCProfile）

定义	
void getTCProfile(in gloabldefs::NamingAttributes_T tcProfileName, out TCProfile_T tcProfile,) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询指定的 TC 模板信息
输入参数	tcProfileName：TC 模板名称
输入/输出参数	无
输出参数	tcProfile：TC 模板信息
操作异常	EXCPT_INTERNAL_ERROR EXCPT_NOT_IMPLEMENTED EXCPT_NE_COMM_LOSS EXCPT_INVALID_INPUT

5.1.14.6.3 查询 TC 模板分配的对象（getTCProfileAssociatedTPs）

定义	
void getTCProfileAssociatedTPs(in gloabldefs::NamingAttributes_T tcProfileName, in unsigned long how_many, out terminationPoint::TerminationPointList_T tpList, out terminationPoint::TerminationPointIterator_I tpIt,) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询 TC 模板分配的对象
输入参数	tcProfileName：TC 模板名称。 how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	tpList：分配的对象列表。 tpIt：迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_NOT_IMPLEMENTED EXCPT_NE_COMM_LOSS EXCPT_INVALID_INPUT EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.14.6.4 查询 TC 模板分配的业务对象（getTCProfileAssociatedBusinesses）

为兼容TMF标准，同时满足本标准第三部分的需求扩充此节。

定义	
void getTCProfileAssociatedBusinesses(in gloabldefs::NamingAttributes_T tcProfileName, in unsigned long how_many, out globaldefs::NameAttributesList_T fdfrNameList,) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询 TC 模板分配的对象
输入参数	tcProfileName：TC 模板名称。 how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	fdfrNameList：分配的业务名称列表
操作异常	EXCPT_INTERNAL_ERROR EXCPT_NOT_IMPLEMENTED EXCPT_NE_COMM_LOSS EXCPT_INVALID_INPUT EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.14.6.5 创建 TC 模板（createTCProfile）

定义	
void createTCProfile(in TCProfileCreateData_T newTCProfileCreateData, out TCProfile_T newTCProfile,) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	创建 TC 模板
输入参数	newTCProfileCreateData：TC 模板创建需要的数据信息
输入/输出参数	无
输出参数	nweTCProfile：创建的 TC 模板信息
操作异常	EXCPT_INTERNAL_ERROR EXCPT_NOT_IMPLEMENTED EXCPT_INVALID_INPUT EXCPT_NE_COMM_LOSS

5.1.14.6.6 删除 TC 模板（deleteTCProfile）

定义	
void deleteTCProfile(in globaldefs::NamingAttributes_T tcProfileName) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	删除指定的 TC 模板
输入参数	tcProfileName：TC 模板名称
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR EXCPT_NOT_IMPLEMENTED EXCPT_OBJECT_IN_USE EXCPT_INVALID_INPUT

5.1.14.6.7 修改 TC 模板（modifyTCProfile）

定义	
void modifyTCProfile(in globaldefs::NamingAttributes_T tcProfileName, in TCProfileCreateData_T tcProfileModifyData, inout subnetworkConnection::TPDataList_T tpsToModify, out TCProfile_T modifiedTCProfile, out string errorReason) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	修改指定的 TC 模板
输入参数	tcProfileName：待修改的 TC 模板名称。 tcProfileModifyData：需要修改的 TC 模板的数据信息

说明	
输入/输出参数	tpsToModify：待修改的 TC 模板相关的 TP 点
输出参数	modifiedTCProfile：修改后的 TC 模板信息。 errorReason：错误原因
操作异常	EXCPT_INTERNAL_ERROR EXCPT_NOT_IMPLEMENTED EXCPT_USERLABEL_IN_USE EXCPT_INVALID_INPUT EXCPT_NE_COMM_LOSS

5.1.14.6.8 分配 TC 模板（assignEXTCProfile）（可选）

定义	
void assignEXTCProfile(in TCProfileAssignTask_T profileTask) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	对指定的实体分配 TC 模板
输入参数	profileTask：模板分配任务
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR EXCPT_NOT_IMPLEMENTED EXCPT_USERLABEL_IN_USE EXCPT_INVALID_INPUT EXCPT_NE_COMM_LOSS

5.1.14.6.9 去分配 TC 模板（deAssignEXTCProfile）（可选）

定义	
void deAssignEXTCProfile(in TCProfileAssignTask_T profileTask) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	对指定的实体去分配 TC 模板
输入参数	profileTask：模板分配任务
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR EXCPT_NOT_IMPLEMENTED EXCPT_USERLABEL_IN_USE EXCPT_INVALID_INPUT EXCPT_NE_COMM_LOSS

5.1.15 OAM 管理模块

5.1.15.1 OAM 维护实体组（EXMegEntityData_T）

定义
struct EXMegEntityData_T {globaldefs::NamingAttributes_T megName; string megID; transmissionParameters::LayerRate_T megLayerRate; globaldefs::NamingAttributes_T assignedEntityName; string megLevel; terminationPoint::TerminationPointList_T mepTPList; terminationPoint::TerminationPointList_T mipTPList; globaldefs::NVSList_T additionalInfo; };

说明	
对象说明	OAM 维护实体组结构
类型取值	取值说明
megName	维护实体组名称
megID	维护实体组 ID
megLayerRate	维护实体组层次
assignedEntityName	维护实体组关联的对象
megLevel	维护实体组层次
mepTPList	维护实体组边缘点。若为 ATM 的 OAM，则此处为 ATM 业务源端接入端口标识
mipTPList	维护实体组中间点。若为 ATM 的 OAM，则此处为 ATM 业务宿端接入端口标识
additionalInfo	附加信息。附加信息中可以包含：ATM 业务名称，ATM 业务 OAM 功能使能禁止状态，ATM 帧发送周期等

5.1.15.2 OAM 维护实体组列表（EXMegEntityDataList_T）

定义	
typedef sequence<EXMegEntityData_T> EXMegEntityDataList_T;	
说明	
对象说明	OAM 维护实体组列表

5.1.15.3 OAM 维护实体组创建数据（EXMEGCreateData_T）

定义	
struct EXMEGCreateData_T {transmissionParameters::LayerRate_T megLayerRate; globaldefs::NamingAttributes_T assignedEntityName; string megLevel; terminationPoint::TerminationPointList_T mepTPList; terminationPoint::TerminationPointList_T mipTPList; globaldefs::NVSList_T additionalInfo; };	
说明	
对象说明	OAM 维护实体组结构
类型取值	取值说明
megID	维护实体组 ID
megLayerRate	维护实体组速率层次

说明	
megLevel	维护实体组层次
mepTPList	维护实体组边缘点。若为 ATM 的 OAM，则此处为 ATM 业务源端接入端口标识
mipTPList	维护实体组中间点。创建 VP 层 OAM 时，此参数为可选。若为 ATM 的 OAM，则此处为 ATM 业务宿端接入端口标识
additionalInfo	附加信息。附加信息中可以包含：ATM 业务名称，ATM 业务 OAM 功能使能禁止状态，ATM 帧发送周期等

5.1.15.4 OAM 维护实体组迭代器接口 ExMegEntityIterator_I

5.1.15.4.1 从迭代器中查询数据（next_n）

定义	
boolean next_n(in unsigned long how_many, out ExMegEntityDataList_T megList) raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	从迭代器中查询 *n* 条记录。当 how_many≥迭代器剩余记录数时，系统在数据取完后应该自动销毁该迭代器对象
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	megList：首次查询返回的 MEG 列表
操作异常	EXCPT_INTERNAL_ERROR

5.1.15.4.2 查询迭代器记录条数（getLength）

定义	
unsigned long getLength() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	查询迭代器中数据记录的总条数（注意这里指的是迭代器中数据记录的总条数，该值在迭代器生命周期内是不变的）
输入参数	无
输入/输出参数	无
输出参数	无
返回值	unsigned long：迭代器中包含的总的记录条数
操作异常	EXCPT_INTERNAL_ERROR

5.1.15.4.3 销毁迭代器对象（destory）

定义	
void destroy() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	销毁迭代器对象
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR

5.1.15.5 OAM 管理接口

5.1.15.5.1 创建 OAM 维护实体组（createMEG）

此节为可选，也可以通过建FDFr时自动创建。

定义	
void createMEG(in exMEGCreateData_T megCreateData, out globaldefs::NamingAttributes_T megName) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	创建 OAM 维护实体组。注意：ATM 层的创建 OAM 维护实体组功能为可选
输入参数	MegCreateData：OAM 维护实体组创建数据
输入/输出参数	无
输出参数	megName：创建后的 MEG 名称信息
操作异常	EXCPT_INTERNAL_ERROR EXCPT_NOT_IMPLEMENTED EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.15.5.2 查询指定对象下的所有 OAM 维护实体组（getAllMEGs）

定义	
void getAllMEGs (in globaldefs::NamingAttributes_T targetObjectName, in unsigned long how_many, out ExMegEntityDataList_T megList, out ExMegEntityIterator_I megIt) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询指定对象下的所有的维护实体组
输入参数	targetObjectName：指定待查询的对象名称。例如：EMS 或 FDFr 名称。 how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	megList：维护实体组列表。 megIt：迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_NOT_IMPLEMENTED EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.15.5.3 查询指定的 OAM 维护实体组（getMEG）

定义
void getAllMEGs (in globaldefs::NamingAttributes_T megName, out ExMegEntityData_T megInfo) raises(globaldefs::ProcessingFailureException);

说明	
功能描述	查询指定的维护实体组
输入参数	megName：指定待查询的维护实体组名称
输入/输出参数	无
输出参数	megInfo：查询到的 MEG 信息
操作异常	EXCPT_INTERNAL_ERROR EXCPT_NOT_IMPLEMENTED EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.1.15.5.4 删除 OAM 维护实体组（deleteMEG）

定义
void deleteMEG (in globaldefs::NamingAttributes_T megName) raises(globaldefs::ProcessingFailureException);

说明	
功能描述	删除指定的维护实体组
输入参数	megName：指定的 OAM 维护实体组名称
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR EXCPT_NOT_IMPLEMENTED EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.2 故障管理模块

PTN新增的故障数据，应符合《分组传送网（PTN）网络管理技术要求 第2部分：NMS功能》相应章节的要求。故障数据应满足本节要求。

5.2.1 告警上报

告警上报数据类型见公共管理部分的通知管理模块。

5.2.2 告警级别模块（module aSAP）

5.2.2.1 分配级别（AssignedSeverity_T）

定义

enum AssignedSeverity_T {
AS_INDETERMINATE,
AS_CRITICAL,
AS_MAJOR,
AS_MINOR,
AS_WARNING,
AS_NONALARMED,
AS_FREE_CHOICE
};

说明	
对象说明	分配级别
类型取值	取值说明
AS_INDETERMINATE	未确认
AS_CRITICAL	严重告警
AS_MAJOR	主要告警
AS_MINOR	次要告警
AS_WARNING	警告
AS_NONALARMED	无告警产生
AS_FREE_CHOICE	原始告警级别

5.2.2.2 告警级别分配（AlarmSeverityAssignment_T）

定义

struct AlarmSeverityAssignment_T {
notifications::ProbableCauseList_T probableCause;
string probableCauseQualifier;
string nativeProbableCause;
AssignedSeverity_T serviceAffecting;
AssignedSeverity_T serviceNonAffecting;
AssignedSeverity_T serviceIndependentOrUnknown;
};

说明	
对象说明	告警级别分配
属性名	属性说明
probableCause	告警原因列表
probableCauseQualifier	告警原因的唯一名称，当告警原因不能保证唯一性时，用来唯一标识告警级别信息
nativeProbableCause	告警在 EMS 本地的原因，当告警原因不能保证唯一性时，也可以用来唯一标识告警级别信息
serviceAffecting	影响服务
serviceNonAffecting	不影响服务
serviceIndependentOrUnknown	未知

5.2.2.3 告警级别分配列表（AlarmSeverityAssignmentList_T）

定义
typedef sequence<AlarmSeverityAssignment_T> AlarmSeverityAssignmentList_T;

说明	
对象说明	告警级别分配信息列表

5.2.2.4 告警级别分配模板（ASAP_T）(可选)

定义
struct ASAP_T { globaldefs::NamingAttributes_T name; string userLabel;

定义
string nativeEMSName; string owner; boolean notModifiable; AlarmSeverityAssignmentList_T alarmSeverityAssignmentList; globaldefs::NVSList_T additionalInfo; };

说明	
对象说明	告警级别分配模板
属性名	属性说明
name	告警级别分配模板名称
userLabel	告警级别分配模板的用户标签
nativeEMSName	告警级别分配模板的本地 EMS 名称
owner	告警级别分配模板的所有者名称
notModifiable	是否不可修改。取值为“真”，表示告警级别是固定的，不能修改
alarmSeverityAssignmentList	告警级别分配列表
additionalInfo	附加信息

5.2.2.5 告警级别分配模板列表（ASAPList_T）(可选)

定义
typedef sequence<ASAP_T> ASAPList_T;

说明	
对象说明	告警级别分配模板列表

5.2.2.6 告警级别分配创建和修改数据（ASAPCreateModifyData_T）（可选）

定义

struct ASAPCreateModifyData_T {
string userLabel;
boolean forceUniqueness;
string owner;
AlarmSeverityAssignmentList_T alarmSeverityAssignmentList;
globaldefs::NVSList_T additionalInfo;
};

说明	
对象说明	告警级别分配创建和修改数据
属性名	属性说明
userLabel	告警级别分配表的用户标签
forceUniqueness	是否要求用户标签唯一
owner	告警级别分配表的所有者
alarmSeverityAssignmentList	告警级别分配列表
additionalInfo	附加信息

5.2.2.7 ASAPIterator_I 接口(可选)

5.2.2.7.1 从迭代器中查询数据（next_n）

定义	
boolean next_n(in unsigned long how_many, out ASAPList_T aSAPList) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	从迭代器中查询 *n* 条记录。当 how_many≥迭代器剩余记录数时，系统在数据取完后应该自动销毁该迭代器对象
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	aSAPList：首次查询返回的告警级别分配模板列表
操作异常	EXCPT_INTERNAL_ERROR

5.2.2.7.2 查询迭代器记录条数（getLength）

定义	
unsigned long getLength() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	查询迭代器中数据记录的总条数（注意这里指的是迭代器中数据记录的总条数，该值在迭代器生命周期内是不变的）
输入参数	无
输入/输出参数	无
输出参数	无
返回值	unsigned long：迭代器中包含的总的记录条数
操作异常	EXCPT_INTERNAL_ERROR

5.2.2.7.3 销毁迭代器对象（destroy）

定义	
void destroy() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	销毁迭代器对象
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR

5.3 性能管理模块

5.3.1 性能管理模块（module performance）

5.3.1.1 性能采集粒度（Granularity_T）

定义	
typedef string Granularity_T;	
说明	
对象说明	表示性能数据采集的粒度周期。 有效值是： "15min"：表示粒度周期为 15min； "24h"：表示粒度周期为 24h； "NA"：表示当前的即时值

5.3.1.2 性能采集粒度周期列表（GranularityList_T）

定义	
typedef sequence <Granularity_T> GranularityList_T;	
说明	
对象说明	性能采集粒度周期列表

5.3.1.3 性能参数名称（PMParameterName_T）

定义	
typedef string PMParameterName_T;	
说明	
对象说明	性能参数名称

5.3.1.4 性能参数名称列表（PMParameterNameList_T）

定义	
typedef sequence<PMParameterName_T> PMParameterNameList_T;	
说明	
对象说明	性能参数名称列表

5.3.1.5 性能测量参数（PMMeasurement_T）

定义	

struct PMMeasurement_T {
PMParameterName_T pmParameterName;
PMLocation_T pmLocation;
float value;
string unit;
string intervalStatus;
};	

说明	
对象说明	性能测量参数
属性名	属性说明
pmParameterName	性能参数名称
pmLocation	性能监测点位置
value	性能值
unit	性能值的单位
intervalStatus	表示描述性能值的有效性，允许的值有： "Valid"：表示数据有效； "Incomplete"：表示整个采集间隔(15min、24h)内，数据不可用； "Invalid"：表示指定的时间间隔内，数据可用但标记为无效； "Unavailable"：表示指定的时间间隔内，数据完全不可用； "Zero-suppressed"：表示零抑制时间段

5.3.1.6 性能测量参数列表（PMMeasurementList_T）

定义	
typedef sequence <PMMeasurement_T> PMMeasurementList_T;	
说明	
对象说明	性能测量参数列表

5.3.1.7 性能数据（PMData_T）

定义

struct PMData_T {
globaldefs::NamingAttributes_T tpName;
transmissionParameters::LayerRate_T layerRate;
Granularity_T granularity;
globaldefs::Time_T retrievalTime;
PMMeasurementList_T pmMeasurementList;
};

说明	
对象说明	性能数据
属性名	属性说明
tpName	终端点名称
layerRate	层速率
granularity	性能采集粒度
retrievalTime	性能查询时间
pmMeasurementList	性能测量参数列表

5.3.1.8 性能数据列表（PMDataList_T）

定义
typedef sequence<PMData_T> PMDataList_T;

说明	
对象说明	性能数据列表

5.3.1.9 性能监测点位置（PMLocation_T）

定义
typedef string PMLocation_T;

说明	
对象说明	表示性能监测的位置。 取值为： "PML_NEAR_END_Rx"：表示近端接收； "PML_FAR_END_Rx"：表示远端接收； "PML_NEAR_END_Tx"：表示近端发送； "PML_FAR_END_Tx"：表示远端发送； "PML_BIDIRECTIONAL"：表示双向； "PML_CONTRA_NEAR_END_Rx"：表示反向的近端接收； "PML_CONTRA_FAR_END_Rx"：表示反向的远端接收

5.3.1.10 性能监测点位置列表（PMLocationList_T）

定义
typedef sequence < PMLocation_T > PMLocationList_T;

说明	
对象说明	性能监测位置列表

5.3.1.11 性能参数（PMParameter_T)

定义

struct PMParameter_T {
PMParameterName_T pmParameterName;
PMLocation_T pmLocation;
};

说明
对象说明	性能参数
属性名	属性说明
pmParameterName	性能参数名称
pmLocation	性能监测点位置

5.3.1.12 性能参数列表（PMParameterList_T)

定义
typedef sequence <PMParameter_T> PMParameterList_T ;
说明
对象说明	性能参数列表

5.3.1.13 性能门限类型（PMThresholdType_T）

定义

enum PMThresholdType_T {
TWM_HIGHEST,
TWM_HIGH,
TWM_LOW,
TWM_LOWEST
};

说明
对象说明	性能门限类型
类型取值	取值说明
TWM_HIGHEST	最高限
TWM_HIGH	高限
TWM_LOW	低限
TWM_LOWEST	最低限

5.3.1.14 性能越限参数（TCAParameter_T）

定义

struct TCAParameter_T {
PMParameterName_T pmParameterName;
Granularity_T granularity;
PMLocation_T pmLocation;
PMThresholdType_T thresholdType;
boolean triggerFlag;
float value;
string unit;
};

说明	
对象说明	性能越限参数
属性名	属性说明
pmParameterName	性能参数名称
granularity	性能采集粒度
pmLocation	性能监测点位置
thresholdType	性能门限类型
triggerFlag	激活标志
value	性能值
unit	性能值的单位

5.3.1.15 性能选择参数（PMTPSelect_T）

定义

struct PMTPSelect_T {
globaldefs::NamingAttributes_T name;
transmissionParameters::LayerRateList_T layerRateList;
PMLocationList_T pMLocationList;
GranularityList_T granularityList;
};

说明	
对象说明	性能选择参数
属性名	属性说明
name	对象名称
layerRateList	速率层次列表
pMLocationList	性能监测点位置列表
granularityList	性能采集粒度周期列表

5.3.1.16 性能选择参数列表（PMTPSelectList_T）

定义	
typedef sequence<PMTPSelect_T> PMTPSelectList_T;	
说明	
对象说明	性能选择参数列表

5.3.1.17 性能门限值（PMThreshold_T）

定义

struct PMThreshold_T {
PMParameterName_T pmParameterName;
PMLocation_T pmLocation;
PMThresholdType_T thresholdType;
boolean triggerFlag;
float value;
string unit;
};

说明	
对象说明	性能门限值
属性名	属性说明
pmParameterName	性能参数名称
pmLocation	性能监测点位置
thresholdType	性能门限类型
triggerFlag	激活标志
value	性能值
unit	性能值的单位

5.3.1.18 性能门限值列表（PMThresholdList_T）

定义	
typedef sequence < PMThreshold_T > PMThresholdList_T;	
说明	
对象说明	性能门限值列表

5.3.1.19 性能保持时间（HoldingTime_T）

定义	

struct HoldingTime_T {
short storeTime24hr;
short storeTime15min;
};	

说明	
对象说明	性能保持时间
属性名	属性说明
storeTime24hr	24h 保持时间
storeTime15min	15min 保持时间

5.3.1.20 性能参数相关的门限（PMParameterWithThresholds_T）

定义	

struct PMParameterWithThresholds_T {
PMParameterName_T pmParameterName;
PMThresholdList_T pmThresholdList;
};	

说明	
对象说明	性能参数相关的门限
属性名	属性说明
pmParameterName	性能参数名称
pmThresholdList	性能门限列表

5.3.1.21 性能参数相关门限的列表（PMParameterWithThresholdsList_T）

定义	
typedef sequence <PMParameterWithThresholds_T> PMParameterWithThresholdsList_T;	
说明	
对象说明	性能参数相关门限的列表

5.3.1.22 管理状态（AdministrativeState_T）

定义

enum AdministrativeState_T {
AS_Locked,
AS_Unlocked
};

说明

对象说明	管理状态
类型取值	取值说明
AS_Locked	锁定
AS_Unlocked	激活

5.3.1.23 性能数据文件目的地址（Destination_T）

定义
typedef string Destination_T;
说明

对象说明	性能数据文件目的地址

5.3.1.24 性能采集任务（PMCollectionTask_T）

定义
struct PMCollectionTask_T { globaldefs::NamingAttributes_T pmCollectionTaskName; string userLabel; string nativeEMSName; string owner; globaldefs::NamingAttributesList_T targetObjectNameList; globaldefs::Time_T collectionBeginTime; globaldefs::Time_T collectionEndTime; Granularity_T granularity; string timeInternal; PMParameterList_T pmParameterList; globaldefs::NVSList_T additionalInfo; };
说明

对象说明	性能采集任务
属性名	属性说明
pmCollectionTaskName	性能采集任务名称
userLabel	性能采集任务友好名称
nativeEMSName	性能采集任务本地名称
owner	性能采集任务所有者
targetObjectName	被采集对象标识符或确定被采集对象的条件
collectionBeginTime	采集起始时间。可选，若不指定，表示立即开始采集
collectionEndTime	性能监测点的本地名称。可选，若不指定，表示一直采集
granularity	采集时间粒度，15min/24h
timeInternal	采集时间间隔
pmParameterList	采集的性能参数列表
additionalInfo	附加信息

5.3.1.25 性能采集任务列表（PMCollectionTaskList_T）

定义	
typedef sequence <PMCollectionTask_T> PMCollectionTaskList_T;	
说明	
对象说明	性能参数相关门限的列表

5.3.1.26 性能采集任务创建结构（PMCollectionTaskCreateData_T）

定义	
struct PMCollectionTaskCreateData_T { string userLabel; string nativeEMSName; string owner; globaldefs::NamingAttributesList_T tartetObjectNameList; globaldefs::Time_T collectionBeginTime; globaldefs::Time_T collectionEndTime; Granularity_T granularity; string timeInternal; PMParameterList_T pmParameterList; globaldefs::NVSList_T additionalInfo; };	
说明	
对象说明	性能采集任务
属性名	属性说明
userLabel	性能采集任务友好名称
nativeEMSName	性能采集任务本地名称
owner	性能采集任务所有者
tartetObjectName	被采集对象标识符或确定被采集对象的条件
collectionBeginTime	采集起始时间。可选，若不指定，表示立即开始采集
collectionEndTime	性能监测点的本地名称。可选，若不指定，表示一直采集
granularity	采集时间粒度，15min/24h
timeInternal	采集时间间隔
pmParameterList	采集的性能参数列表。可选，若不指定，表示所有数据都采集
additionalInfo	附加信息

5.3.1.27 PMDataIterator_I 接口

5.3.1.27.1 从迭代器中查询数据（next_n）

定义	
boolean next_n(in unsigned long how_many, out PMDataList_T pmDataList) raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	从迭代器中查询 *n* 条记录。当 how_many≥迭代器剩余记录数时，系统在数据取完后应该自动销毁该迭代器对象
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	pmDataList：首次查询返回的性能数据列表
操作异常	EXCPT_INTERNAL_ERROR

5.3.1.27.2 查询迭代器记录条数（getLength）

定义	
unsigned long getLength() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	查询迭代器中数据记录的总条数（注意这里指的是迭代器中数据记录的总条数，该值在迭代器生命周期内是不变的）
输入参数	无
输入/输出参数	无
输出参数	无
返回值	unsigned long：迭代器中包含的总的记录条数
操作异常	EXCPT_INTERNAL_ERROR

5.3.1.27.3 销毁迭代器对象（destroy）

定义	
void destroy() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	销毁迭代器对象
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR

5.3.1.28 PMPIterator_I 接口

5.3.1.28.1 从迭代器中查询数据（next_n）

定义	
boolean next_n(in unsigned long how_many, out PMPList_T pmpList) raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	从迭代器中查询 *n* 条记录。当 how_many≥迭代器剩余记录数时，系统在数据取完后应该自动销毁该迭代器对象
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	pmpList：首次查询返回的性能监测点列表
操作异常	EXCPT_INTERNAL_ERROR

5.3.1.28.2 查询迭代器记录条数（getLength）

定义	
unsigned long getLength() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	查询迭代器中数据记录的总条数（注意这里指的是迭代器中数据记录的总条数，该值在迭代器生命周期内是不变的）

说明	
输入参数	无
输入/输出参数	无
输出参数	无
返回值	unsigned long：迭代器中包含的总的记录条数
操作异常	EXCPT_INTERNAL_ERROR

5.3.1.28.3 销毁迭代器对象（destroy）

定义	
void destroy() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	销毁迭代器对象
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR

5.3.1.29 PerformanceManagementMgr_I 接口

5.3.1.29.1 查询网元性能的能力（getMEPMCapabilities）

定义	
void getMEPMcapabilities(in globaldefs::NamingAttributes_T meName, in transmissionParameters::LayerRate_T layerRate, out PMParameterList_T pmParameterList) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	指定对象名称，查询网元性能的能力
输入参数	meName：指定的网元名称。 layerRate：指定的速率层次
输入/输出参数	无
输出参数	pmParameterList：性能参数
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_NE_COMM_LOSS EXCPT_CAPACITY_EXCEEDED

5.3.1.29.2 激活性能采集数据（enablePMData）

定义
void enablePMData(in PMTPSelectList_T pmTPSelectList, out PMTPSelectList_T failedTPSelectList) raises(globaldefs::ProcessingFailureException);

说明	
功能描述	激活指定性能监测点上的性能采集
输入参数	pmTPSelectList：性能选择参数列表
输入/输出参数	无
输出参数	failedTPSelectList：激活失败的性能选择参数列表
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_NE_COMM_LOSS EXCPT_CAPACITY_EXCEEDED

5.3.1.29.3 去激活性能采集数据（disablePMData）

定义	
void disablePMData(in PMTPSelectList_T pmTPSelectList, out PMTPSelectList_T failedTPSelectList) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	去激活指定性能监测点上的性能采集
输入参数	pmTPSelectList：性能选择参数列表
输入/输出参数	无
输出参数	failedTPSelectList：去激活失败的性能选择参数列表
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_NE_COMM_LOSS

5.3.1.29.4 清除性能寄存器（clearPMData）

定义	
void clearPMData(in PMTPSelectList_T pmTPSelectList, out PMTPSelectList_T failedTPSelectList) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	指定对象名称，清除该对象的性能寄存器
输入参数	pmTPSelectList：性能选择参数列表
输入/输出参数	无
输出参数	failedTPSelectList：失败的性能选择参数列表
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_NE_COMM_LOSS

5.3.1.29.5 查询所有的当前性能数据（getAllCurrentPMData）

定义
void getAllCurrentPMData(in PMTPSelectList_T pmTPSelectList, in PMParameterNameList_T pmParameters,

定义	
in unsigned long how_many, out PMDataList_T pmDataList, out PMDataIterator_I pmIt) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	指定对象名称和性能参数，查询所有的当前性能
输入参数	pmTPSelectList：性能选择参数列表。 pmParameters：性能参数。 how_many：迭代查询数据方式下首次查询的数目
输入/输出参数	无
输出参数	pmDataList：性能数据列表。 pmIt：迭代器
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_NE_COMM_LOSS EXCPT_UNABLE_TO_COMPLY

5.3.1.29.6 查询历史性能数据（getHistoryPMData）

定义	
void getHistoryPMData (in Destination_T destination, in string userName, in string password, in PMTPSelectList_T pmTPSelectList, in PMParameterNameList_T pmParameters, in globaldefs::Time_T startTime, in globaldefs::Time_T endTime, in boolean forceUpload) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	指定对象名称和性能参数，将历史性能文件上载到指定的文件传输服务器
输入参数	destination：文件传输服务器目标地址。 userName：文件传输用户名。 password：文件传输密码。 pmTPSelectList：性能选择参数列表。 pmParameters：性能参数。 startTime：历史性能起始时间。 endTime：历史性能结束时间。 forceUpload：是否强制全部上载
输入/输出参数	无
输出参数	无
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_NE_COMM_LOSS EXCPT_UNABLE_TO_COMPLY

5.3.1.29.7 查询保持时间（getHoldingTime）

定义	
void getHoldingTime(out HoldingTime_T holdingTime) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询 EMS 的性能保持时间
输入参数	无
输入/输出参数	无
输出参数	holdingTime：性能保持时间
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_NE_COMM_LOSS

5.3.1.29.8 激活性能越限告警上报（enableTCA）

定义	
void enableTCA(in PMTPSelectList_T pmTPSelectList, out PMTPSelectList_T failedTPSelectList) raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	激活性能越限告警上报
输入参数	pmTPSelectList：性能选择参数列表
输入/输出参数	无
输出参数	failedTPSelectList：失败的性能选择参数列表
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_NE_COMM_LOSS EXCPT_UNABLE_TO_COMPLY

5.3.1.29.9 去激活性能越限告警上报（disableTCA）

定义	
void disableTCA(in PMTPSelectList_T pmTPSelectList, out PMTPSelectList_T failedTPSelectList) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	去激活性能越限告警上报
输入参数	pmTPSelectList：性能选择参数列表
输入/输出参数	无
输出参数	failedTPSelectList：失败的性能选择参数列表
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.3.1.29.10 设置监测点的性能门限值（setTCATPParameter）

定义	
void setTCATPParameter(in NamingAttributes_T tpName, inout TCAParameter_T tcaParameters) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	设置监测点的性能越限门限值
输入参数	tpName：监测点名称
输入/输出参数	tcaParameter：性能越限参数值
输出参数	无
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.3.1.29.11 查询监测点的性能门限值（getTCATPParameter）

定义	
void getTCATPParameter(in NamingAttributes_T tpName, in LayerRate_T layerRate, in Granularity_T granularity, out TCAParameter_T tcaParameter) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询监测点的性能越限门限值
输入参数	tpName：监测点名称。 layerRate：监测的速率层次。 granularity：监测力度
输入/输出参数	无
输出参数	tcaParameter：性能越限参数值
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.3.1.29.12 查询终端点的历史性能数据（getTPHistoryPMData）

定义
void getTPHistoryPMData(in PMTPSelectList_T pmTPSelectList, in PMParameterNameList_T pmParameters, in globaldefs::Time_T startTime, in globaldefs::Time_T endTime,

定义	
in unsigned long how_many, out PMDataList_T pmDataList, out PMDataIterator_I pmIt) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询指定终端点、时间段的历史性能数据
输入参数	pmTPSelectList：性能选择参数列表。 pmParameters：性能参数。 startTime：开始时间。 endTime：结束时间。 how_many：迭代查询数据方式下首次查询的数目
输入/输出参数	无
输出参数	pmDataList：性能数据列表。 pmIt：迭代器
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_UNABLE_TO_COMPLY EXCPT_INVALID_INPUT EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

5.3.1.29.13 查询性能采集任务（getAllPMCollectionTasks）

定义	
void getAllPMCollectionTasks(in globaldefs::NamingAttributes_T emsName, out PMCollectionTaskList_T pmCollectionTaskList) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	指定对象名称，查询全部性能采集任务
输入参数	emsName：指定的 EMS 对象名称
输入/输出参数	无
输出参数	pmCollectionTaskList：性能采集任务列表
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.3.1.29.14 删除性能采集任务（deletePMCollectionTask）

定义	
void deletePMCollectionTask(in globaldefs::NamingAttributes_T targetObjectName) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	指定对象删除性能采集任务
输入参数	targetObjectName：指定对象名称，删除该指定对象上的性能采集任务
输入/输出参数	无
输出参数	thePMCollectionTask：被删除的性能采集任务
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.3.1.29.15 创建性能采集任务（createPMCollectionTask）

定义	
void createPMCollectionTask(in PMCollectionTaskCreateData_T pmCollectionTaskCreateData, out PMCollectionTask_T newPMCollectionTask) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	创建性能采集任务
输入参数	pmCollectionTaskCreateData：待创建的性能采集任务信息
输入/输出参数	无
输出参数	newPMCollectionTask：创建后的性能采集任务
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.3.1.29.16 修改性能采集任务（modifyPMCollectionTask）

定义	
void modifyPMCollectionTask(in PMCollectionTask_T toModifyPMColTaskData out PMCollectionTask_T thePMCollectionTask) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	修改性能采集任务
输入参数	toModifyPMColTaskData：指定要修改的性能采集任务信息
输入/输出参数	无
输出参数	thePMCollectionTask：修改后的性能采集任务信息

说明	
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.3.1.29.17 暂停性能采集任务（pausePMCollectionTask）

定义	
void pausePMCollectionTask(in globaldefs::NamingAttributes_T targetObjectName) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	暂停指定对象上的性能采集任务
输入参数	targetObjectName：指定对象名称，例如网元、TP 点等
输入/输出参数	无
输出参数	无
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.3.1.29.18 恢复性能采集任务（resumePMCollectionTask）

定义	
void resumePMCollectionTask(in globaldefs::NamingAttributes_T targetObjectName) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	恢复指定对象上的性能采集任务
输入参数	targetObjectName：指定对象名称，例如网元、TP 点等
输入/输出参数	无
输出参数	无
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.3.1.29.19 获取历史性能数据（getHistoryPMDataByPull）

定义
void getHistoryPMDataByPull (in string taskName,

定义
in string compressType, in string packingType, in PMTPSelectList_T pmTPSelectList, in PMParameterNameList_T pmParameters, in globaldefs::Time_T startTime, in globaldefs::Time_T endTime) raises(globaldefs::ProcessingFailureException);

说明	
功能描述	获取历史性能数据
输入参数	taskName：NMS 指定的任务名称，由 NMS 自定义并保证唯一性 compressType：指定文件是否压缩以及压缩包的类型（取值为 NO_COMPRESSION、GZIP）” packingType：指定文件是否打包以及打包的方式（取值为 NO_PACKING、ZIP、TAR）” pmTPSelectList：指定查询的性能监测点的对象列表。包括性能监视对象的标识符，如指定的端口、终端点等；层速率列表；时间粒度（15min 或 24h）；性能测量位置信息列表；起始时间和终止时间。 pmParameters：指定需要查询的性能参数列表，若为空，则查询性能监视对象的所有性能参数。 startTime：指定查询的起始时间（include）。 endTime：指定查询的终止时间（exclude）
输入/输出参数	无
输出参数	全网历史告警文件命名见附录 B。 历史告警文件的保留时间为 3 天（72h）
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.4 L3VPN 配置管理模块

5.4.1 L3VPN 配置管理模块

5.4.1.1.1 二三层桥接配置对象(Bridge_T)

定义

struct Bridge_T
{
globaldefs::NamingAttributes_T name;
string userLabel;
string nativeEMSName;
string owner;
globaldefs::ConnectionDirection_T direction;
globaldefs::NamingAttributesList_T aEnd;
globaldefs::NamingAttributesList_T zEnd;
transmissionParameters::LayeredParameterList_T transmissionParams;
globaldefs::NVSList_T additionalInfo;
};

说明	
对象说明	性能选择参数
属性名	属性说明
name	对象名称
userLabel	
nativeEMSName	
owner	
direction	
aEnd	
zEnd	
transmissionParams	
additionalInfo	

5.4.1.1.2 VRF 配置对象(VRF_T)

定义

struct VRF_T
{
globaldefs::NamingAttributes_T name;
string userLabel;
string nativeEMSName;
string owner;
string vrfType;
string vrfLabel;
subnetworkConnection::TPDataList_T aEnd;
subnetworkConnection::TPDataList_T zEnd;
string rdFormat;
string rdAttr;
ImportRTList_T importRTList;
ExportRTList_T exportRTList;
transmissionParameters::LayeredParameterList_T transmissionParams;
globaldefs::NVSList_T additionalInfo;
};

说明	
对象说明	性能选择参数
属性名	属性说明
name	对象名称
userLabel	
nativeEMSName	
owner	
vrfType	
vrfLabel	
aEnd	
zEnd	
rdFormat	

说明	
rdAttr	
importRTList	
exportRTList	
transmissionParams	
additionalInfo	

5.4.1.1.3 FRR 保护对象(FRRProtection_T)

定义

struct FRRProtection_T
{
globaldefs::NamingAttributes_T name;
string userLabel;
string nativeEMSName;
string owner;
string protectionType;
transmissionParameters::LayerRate_T rate;
globaldefs::NamingAttributesList_T tpList;
globaldefs::NVSList_T frrParameters;
globaldefs::NamingAttributes_T bindingObject;
globaldefs::NVSList_T additionalInfo;
};

说明	
对象说明	性能选择参数
属性名	属性说明
name	对象名称
userLabel	
nativeEMSName	
owner	
protectionType	
rate	
tpList	
frrParameters	
bindingObject	
additionalInfo	

5.4.1.1.4 VRRP 保护对象(VRRPProtection_T)

定义
struct VRRPProtection_T { globaldefs::NamingAttributes_T name; string userLabel; string nativeEMSName; string owner; transmissionParameters::LayerRate_T rate;

定义
globaldefs::NamingAttributesList_T tpList; globaldefs::NVSList_T vrrpParameters; globaldefs::NVSList_T additionalInfo; };

说明	
对象说明	性能选择参数
属性名	属性说明
name	对象名称
userLabel	
nativeEMSName	
owner	
rate	
tpList	
vrrpParameters	
additionalInfo	

5.4.1.1.5 静态路由表对象(StaticRouting_T)

定义

struct StaticRouting_T
{
string destIP;
string destMask;
string nextHopIP;
globaldefs::NamingAttributes_T outPort;
string priority;
globaldefs::NamingAttributes_T bindingObject;
globaldefs::NVSList_T additionalInfo;
};

说明	
对象说明	性能选择参数
属性名	属性说明
destIP	对象名称
destMask	
nextHopIP	
outPort	
priority	
bindingObject	
additionalInfo	

5.4.2 IPMgr_I 接口

5.4.2.1.1 获取指定网元上的所有二三层端口桥接配置(getAllBridges)

定义
void getAllBridges(in globaldefs::NamingAttributes_T managedElementName,

定义	
in unsigned long how_many, out BridgeList_T bridgeList, out BridgeIterator_I bridgeIt) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	指定网元名称，查询指定网元上的所有二三层端口桥接配置
输入参数	managedElementName：网元名称。 how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	bridgeList：二三层端口桥接配置列表。 bridgeIt：迭代器
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.4.2.1.2 获取指定网元上的所有静态路由配置信息(getAllStaticRoutings)

定义	
void getAllStaticRoutings(in globaldefs::NamingAttributes_T managedElementName, in unsigned long how_many, out StaticRoutingList_T srList, out StaticRoutingIterator_I srIt) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	指定网元名称，查询指定网元上的所有静态路由配置信息
输入参数	managedElementName：网元名称。 how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	srList：静态路由配置列表。 srIt：迭代器
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.4.2.1.3 查询指定网元的所有 VRF 配置(getAllVRFs)

定义
void getAllVRFs(in globaldefs::NamingAttributes_T managedElementName,

定义
in unsigned long how_many, out VRFList_T vrfList, out VRFIterator_I vrfIt) raises(globaldefs::ProcessingFailureException);
说明
功能描述	指定网元名称，查询指定网元的所有 VRF 配置
输入参数	managedElementName：网元名称。 how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	vrfList：VRF 配置列表。 vrfIt：迭代器
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.4.2.1.4 查询指定 L3VPN 关联的 VRF 配置(getFDFrVRFs)

定义
void getFDFrVRFs(in globaldefs::NamingAttributes_T fdfrName, in unsigned long how_many, out VRFList_T vrfList, out VRFIterator_I vrfIt) raises(globaldefs::ProcessingFailureException);
说明
功能描述	指定对象名称，查询全部性能采集任务
输入参数	fdfrName：L3VPN 名称。 how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	vrfList：VRF 配置列表。 vrfIt：迭代器
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.4.2.1.5 查询指定网元内的所有 FRR 保护信息(getAllFRRs)

定义
void getAllFRRs(in globaldefs::NamingAttributes_T managedElementName,

定义	
in unsigned long how_many, out FRRList_T frrList, out FRRIterator_I frrIt) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	指定网元名称，查询指定网元内的所有 FRR 保护信息
输入参数	managedElementName：网元名称。 how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	frrList：FRR 保护信息列表。 frrIt：迭代器
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.4.2.1.6 查询指定网元内的所有 VRRP 保护信息(getAllVRRPs)

定义	
void getAllVRRPs(in globaldefs::NamingAttributes_T managedElementName, in unsigned long how_many, out VRRPList_T vrrpList, out VRRPIterator_I vrrpIt) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	指定网元名称，查询指定网元内的所有 VRRP 保护信息
输入参数	managedElementName：网元名称。 how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	vrrpList：VRRP 保护信息列表。 vrrpIt：迭代器
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.5 通用管理模块

5.5.1 会话管理模块（module session）

会话部分具体实现有两个会话接口：一个实现在EMS，即EMSSession_I；另一个实现在NMS，即NMSSession_I。它们都是本接口的派生接口。

5.5.1.1 Session_I 接口

5.5.1.1.1 通信监测操作（ping）

定义	
void ping ();	
说明	
功能描述	探测通信的丢失
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	无

5.5.1.1.2 结束会话（endSession）

定义	
oneway void endSession ( );	
说明	
功能描述	用于结束会话，NMS 和 EMS 都可以发起此操作
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	无

5.5.2 EMS 会话厂管理模块（module emsSessionFactory）

5.5.2.1 EmsSessionFactory_I 接口(从 mtnmVersion::Version_I 继承)

EMS会话厂接口。提供了访问EMS 的入口。通过EMS会话厂可以进一步查询EMS会话接口。

5.5.2.1.1 查询 EMS 会话接口（getEmsSession）

定义	

void getEmsSession(
in string user,
in string password,
in nmsSession::NmsSession_I client,
out emsSession::EmsSession_I emsSessionInterface)
raises(globaldefs::ProcessingFailureException);
};	

说明	
功能描述	操作允许 NMS 查询 EmsSession_I 接口对象，其包含了 EMS 能够包含的管理者对象
输入参数	user：用户名。 password：用户口令。 client：NMS 的会话操作接口，用于 EMS 操作使用，如上报事件等
输入/输出参数	无
输出参数	emsSessionInterface：EMS 会话操作接口
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ACCESS_DENIED

5.5.3 EMS 会话管理模块（module emsSession）

5.5.3.1 EmsSession_I 接口（从 session::Session_I 继承）

EMS会话管理接口，用于提供对EMS访问操作的各管理者。

5.5.3.1.1 管理者名称信息（managerNames_T）

定义	
typedef sequence<string> managerNames_T;	
说明	
对象说明	管理者名称信息。在 NMS 查询 EMS 支持的管理者信息时使用

5.5.3.1.2 查询 EMS 支持的管理者信息（getSupportedManagers）

定义	
void getSupportedManagers(out managerNames_T supportedManagerList) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询 EMS 支持的管理者信息。根据查询到的管理者信息，可以进一步查询管理者的操作接口
输入参数	无
输入/输出参数	无
输出参数	supportedManagerList：管理者名称列表。要求必须一致的管理者名称有： "EMS"：EMS 管理者。 "ManagedElement"：网元管理者。 "MultiLayerSubnetwork" ：子网管理者
操作异常	EXCPT_INTERNAL_ERROR EXCPT_ACCESS_DENIED

5.5.3.1.3 查询管理者操作接口（getManager）

定义	
void getManager(in string managerName, out common::Common_I managerInterface) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	查询 EMS 支持的管理者信息。根据查询到的管理者信息，可以进一步查询管理者的操作接口
输入参数	managerName：管理者名称
输入/输出参数	无
输出参数	managerInterface：管理者操作接口，管理者的操作接口都是从 common 接口继承的
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_ACCESS_DENIED

5.5.3.1.4 查询事件通道（getEventChannel）

定义

void getEventChannel(
out CosNotifyChannelAdmin::EventChannel eventChannel)
raises(globaldefs::ProcessingFailureException);
};

说明	
功能描述	查询 EMS 提供的事件通道。NMS 可以利用此事件通道接收 EMS 发送过来的事件
输入参数	无
输入/输出参数	无
输出参数	eventChannel：EMS 提供的事件通道
操作异常	EXCPT_INTERNAL_ERROR EXCPT_ACCESS_DENIED

5.5.4 NMS 会话管理模块（module nmsSession）

5.5.4.1 NmsSession_I 接口（从 session::Session_I 继承）

NMS 会话管理接口。本接口提供给 EMS 操作使用，在查询 EMS 会话接口时提供。

5.5.4.1.1 通知事件丢失（eventLossOccurred）

定义	
void eventLossOccurred(in globaldefs::Time_T startTime, in string notificationId);	
说明	
功能描述	事件丢失通知。当 EMS 向 NMS 推事件失败时，它会使用此操作向 NMS 发出事件丢失通知
输入参数	startTime：事件丢失的起始时间。 notificationId：通知标识符
输入/输出参数	无
输出参数	无
操作异常	无

5.5.4.1.2 通知事件丢失清除（eventLossCleared）

定义	
void eventLossCleared(in globaldefs::Time_T endTime);	
说明	
功能描述	事件丢失清除通知。当 EMS 向 NMS 推事件由失败恢复正常时，它会使用此操作向 NMS 发出事件丢失清除通知
输入参数	endTime：事件丢失的结束时间
输入/输出参数	无
输出参数	无
操作异常	无

5.5.5 传送参数管理模块（module transmissionParameters）

5.5.5.1 层速率（LayerRate_T）

定义	
typedef short LayerRate_T;	
说明	
对象说明	层速率

5.5.5.2 层速率列表（LayerRateList_T）

定义	
typedef sequence<LayerRate_T> LayerRateList_T;	
说明	
对象说明	层速率列表

5.5.5.3 层速率及其相关的传送参数列表（LayeredParameters_T）

定义	

struct LayeredParameters_T
{
LayerRate_T layer;
globaldefs::NVSList_T transmissionParams;
};	

说明	
对象说明	层速率及其相关的传送参数列表
属性名	属性说明
layer	层速率
transmissionParams	层参数列表

5.5.5.4 层速率及参数列表的列表（LayeredParameterList_T）

定义	
typedef sequence<LayeredParameters_T> LayeredParameterList_T;	
说明	
对象说明	层速率及参数列表的列表

5.5.6 时间同步管理模块（module timeMgr）

5.5.6.1 EMSTimeMgr_I 接口（从 common::Common_I 继承）

时间管理接口。主要提供对EMS时间的操作，如果EMS支持NTP协议，设置时间操作可选。

5.5.6.1.1 查询 EMS 时间（getEMSTime）

定义	
void getEMSTime (out globaldefs::Time_T emsTime) raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	用于 NMS 查询 EMS 时间
输入参数	无
输入/输出参数	无
输出参数	emsTime：EMS 的当前时间
操作异常	EXCPT_INTERNAL_ERROR

5.5.6.1.2 设置 EMS 时间（setEMSTime）(可选)

定义	
void setEMSTime (in globaldefs::Time_T setTime) raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	用于 NMS 设置 EMS 时间，在 EMS 不支持 NTP 协议时使用
输入参数	setTime：设置时间
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR

5.5.7 维护管理模块（module maintenanceOps）

5.5.7.1 维护操作（MaintenanceOperation_T）

定义	
typedef string MaintenanceOperation_T;	
说明	
对象说明	维护操作类型，具体定义如下： "FACILITY_LOOPBACK"：表示向设备侧环回。 "TERMINAL_LOOPBACK"：表示向终端侧环回。 "FACILITY_FORCED_AIS"：表示向设备侧强制插入告警指示信号。 "TERMINAL_FORCED_AIS"：表示向终端侧强制插入告警指示信号。 "FORCE_RDI"：表示强插入远端缺陷指示

5.5.7.2 维护操作模式（MaintenanceOperationMode_T）

定义	

enum MaintenanceOperationMode_T {
MOM_OPERATE,
MOM_RELEASE
};	

说明	
对象说明	维护操作模式
类型取值	取值说明
MOM_OPERATE	执行维护操作
MOM_RELEASE	解除维护操作

5.5.7.3 当前维护操作（CurrentMaintenanceOperation_T）

定义	

struct CurrentMaintenanceOperation_T {
globaldefs::NamingAttributes_T tpName;
MaintenanceOperation_T maintenanceOperation;
transmissionParameters::LayerRate_T layerRate;
globaldefs::NVSList_T additionalInfo;
};	

说明	
对象说明	当前的维护操作
属性名	属性说明
tpName	终端点名称
maintenanceOperation	维护操作类型
layerRate	层速率
additionalInfo	附加信息

5.5.7.4 当前维护操作列表（CurrentMaintenanceOperationList_T）

定义	
typedef sequence<CurrentMaintenanceOperation_T > CurrentMaintenanceOperationList_T;	
说明	
对象说明	当前维护操作列表

5.5.7.5 CurrentMaintenanceOperationIterator_I 接口

5.5.7.5.1 从迭代器中查询数据（next_n）

定义	
boolean next_n(in unsigned long how_many, out CurrentMaintenanceOperationList_T cMOList) raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	从迭代器中查询 *n* 条记录。当 how_many≥迭代器剩余记录数时，系统在数据取完后应该自动销毁该迭代器对象
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	cMOList：首次查询返回的维护操作数据列表
操作异常	EXCPT_INTERNAL_ERROR

5.5.7.5.2 查询迭代器数据记录条数（getLength）

定义	
unsigned long getLength() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	查询迭代器中数据记录的总条数（注意这里指的是迭代器中数据记录的总条数，该值在迭代器生命周期内是不变的）
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR

5.5.7.5.3 销毁迭代器对象（destroy）

定义	
void destroy() aises (globaldefs::ProcessingFailureException);	
说明	
功能描述	销毁迭代器对象
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR

5.5.7.6 MaintenanceMgr_I 接口（从 common::Common_I 继承）

维护操作管理接口。主要提供执行维护操作和获取当前的维护操作数据等。

5.5.7.6.1 执行或解除维护操作（performMaintenanceOperation）

定义
void performMaintenanceOperation(in CurrentMaintenanceOperation_T maintenanceOperation, in MaintenanceOperationMode_T maintenanceOperationMode) raises (globaldefs::ProcessingFailureException);

说明	
功能描述	用于 NMS 向 EMS 下发维护操作命令，可以是执行维护操作或解除已经执行的维护操作
输入参数	maintenanceOperation：当前的维护操作信息。 maintenanceOperationMode：维护操作的方式，包括执行或解除两种
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR EXCPT_NOT_IMPLEMENTED EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS EXCPT_UNABLE_TO_COMPLY EXCPT_NOT_IN_VALID_STATE

5.5.7.6.2 查询当前的维护操作（getActiveMaintenanceOperations）

定义
void getActiveMaintenanceOperations(in globaldefs::NamingAttributes_T tpOrMeName, in unsigned long how_many, out CurrentMaintenanceOperationList_T currentMaintenanceOpeationList, out CurrentMaintenanceOperationIterator_I cmoIt) raises (globaldefs::ProcessingFailureException);

说明	
功能描述	用于 NMS 查询网元或终端点上当前的维护操作信息
输入参数	tpOrMeName：终端点或网元名称。 how_many：迭代查询数据时首次查询的数目
输入/输出参数	无
输出参数	currentMaintenanceOpeationList：当前的维护操作列表。 cmoIt：迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_NOT_IMPLEMENTED EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS EXCPT_UNABLE_TO_COMPLY EXCPT_TOO_MANY_OPEN_ITERATORS

5.5.7.6.3 执行或解除维护操作（performEXMaintenanceOperation）

定义
void performEXMaintenanceOperation(in maintenanceOps::CurrentMaintenanceOperation_T maintenanceOperation, in maintenanceOps::MaintenanceOperationMode_T maintenanceOperationMode, out globaldefs ::NVSList_T resultOfMaintenanceData) raises (globaldefs::ProcessingFailureException);

说明	
功能描述	用于 NMS 向 EMS 下发维护操作命令，可以是执行维护操作或解除已经执行的维护操作。同时返回操作的结果
输入参数	maintenanceOperation：当前的维护操作信息。 maintenanceOperationMode：维护操作的方式，包括执行或解除两种
输入/输出参数	无
输出参数	resultOfMaintenanceData：执行维护操作后的结果
操作异常	EXCPT_INTERNAL_ERROR EXCPT_NOT_IMPLEMENTED EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS EXCPT_UNABLE_TO_COMPLY EXCPT_NOT_IN_VALID_STATE

5.6 公共管理模块

5.6.1 全局定义（module globaldefs）

5.6.1.1 名值对（NameAndStringValue_T)

定义

struct NameAndStringValue_T
{
string name;
string value;
};

说明	
对象说明	NameAndStringValue_T 结构用于替代 OMG 定义的名值对列表（NVList），从性能和开销的角度考虑，将 value 定义为 string 比将其定义为 any 类型更有效，本接口使用元组序列来表示对象的命名
属性名	属性说明
name	对象的名称
value	对象的值

5.6.1.2 名值对列表（NVSList_T）

定义
typedef sequence<NameAndStringValue_T> NVSList_T;

说明	
对象说明	名值对列表

5.6.1.3 对象的名值对（NamingAttributes_T）

定义
typedef NVSList_T NamingAttributes_T;

说明	
对象说明	对象的名值对

5.6.1.4 对象的名值对列表（NamingAttributesList_T）

定义	
typedef sequence<NamingAttributes_T> NamingAttributesList_T;	
说明	
对象说明	对象的名值对列表

5.6.1.5 对象的名值对列表对（NamingAttributesMultipleList_T）

定义	
typedef sequence<globaldefs::NamingAttributesList_T> NamingAttributesMultipleList_T;	
说明	
对象说明	对象的名值对列表对

5.6.1.6 时间（Time_T）

定义	
typedef string Time_T;	
说明	
对象说明	时间。其表示方式遵循 ITU-T X.208 中关于时间表示的定义， 其格式为 "yyyyMMddhhmmss.s[Z|{+|-}HHMm]"，其中： yyyy：年，取值为"0000".."9999"； MM：月，取值为"01".."12"； dd：日，取值为"01".."31"； hh：时，取值为"00".."23"； mm：分，取值为"00".."59"； ss：秒，取值为"00".."59"； Z：表示为 UTC，而不是本地时间，取值为"Z"； {+|-}：与 UTC 时间的时差，取值为"+"或"-"； HH：时差中的小时部分，取值为"00".."23"； Mm：时差中的分钟部分，取值为"00".."59"

5.6.1.7 连接方向（ConnectionDirection_T）

定义	

enum ConnectionDirection_T {
CD_UNI,
CD_BI
};	

说明	
对象说明	连接方向。可表示子网连接、交叉连接、拓扑连接的方向
属性名	属性说明
CD_UNI	单向
CD_BI	双向

5.6.1.8 异常类型(ExceptionType_T)

定义
enum ExceptionType_T { EXCPT_NOT_IMPLEMENTED, EXCPT_INTERNAL_ERROR, EXCPT_INVALID_INPUT,

定义
EXCPT_OBJECT_IN_USE, EXCPT_TP_INVALID_ENDPOINT, EXCPT_ENTITY_NOT_FOUND, EXCPT_TIMESLOT_IN_USE, EXCPT_PROTECTION_EFFORT_NOT_MET, EXCPT_NOT_IN_VALID_STATE, EXCPT_UNABLE_TO_COMPLY, EXCPT_NE_COMM_LOSS, EXCPT_CAPACITY_EXCEEDED, EXCPT_ACCESS_DENIED, EXCPT_TOO_MANY_OPEN_ITERATORS, EXCPT_UNSUPPORTED_ROUTING_CONSTRAINTS, EXCPT_USERLABEL_IN_USE };

说明	
对象说明	在处理出错时抛出的异常类型
属性名	属性说明
EXCPT_NOT_IMPLEMENTED	操作没有完成或不支持操作
EXCPT_INTERNAL_ERROR	EMS 内部处理错误
EXCPT_INVALID_INPUT	无效的输入
EXCPT_OBJECT_IN_USE	对象正被使用中。如配置的终端点已被用作交叉连接或被终结或已被映射，就不能再被用来做有冲突的配置
EXCPT_TP_INVALID_ENDPOINT	指定的终端点不存在或不能被创建
EXCPT_ENTITY_NOT_FOUND,	实体没有找到。如使用 EMS 支持的某个对象名称作为参数来操作某个对象，但又没有找到此对象，就可抛出此异常
EXCPT_TIMESLOT_IN_USE	创建和激活子网连接时涉及的时隙正在被 EMS 所占用
EXCPT_PROTECTION_EFFORT_NOT_MET	NMS 要求 SNC 的保护尽力程度 EMS 不能满足
EXCPT_NOT_IN_VALID_STATE	当前的状态不能进行某一操作。如不能删除处于激活状态的 SNC
EXCPT_UNABLE_TO_COMPLY	在 EMS 不能响应操作请求时抛出
EXCPT_NE_COMM_LOSS	与网元的通信中断
EXCPT_CAPACITY_EXCEEDED	当某一操作请求超出了 EMS 或网元的处理能力时抛出
EXCPT_ACCESS_DENIED	当操作的安全认证失败时抛出
EXCPT_TOO_MANY_OPEN_ITERATORS	当请求迭代操作时总的迭代器数目超出了 EMS 的迭代器数目限制时抛出
EXCPT_UNSUPPORTED_ROUTING_CONSTRAINTS	EMS 不支持设置的路由限制
EXCPT_USERLABEL_IN_USE	用户友好名称正在使用中，即友好名称已存在

5.6.1.9 处理失败时的异常类型（ProcessingFailureException）

定义

exception ProcessingFailureException
{
ExceptionType_T exceptionType;
string errorReason;
};

说明	
对象说明	处理失败抛出的异常
属性名	属性说明
exceptionType	异常类型
errorReason	产生异常的原因。是对前面的 ExceptionType 进行补充说明，对 errorReason 的具体格式没有定义，只要是一个说明出错原因的可读 string 即可

5.6.1.10 NamingAttributesIterator_I 接口

5.6.1.10.1 从迭代器中查询数据（next_n）

定义	
boolean next_n(in unsigned long how_many, out NamingAttributesList_T nameList) raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	从迭代器中查询 *n* 条记录。当 how_many≥迭代器剩余记录数时，系统在数据取完后应该自动销毁该迭代器对象
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	nameList：首次查询返回的名值对列表
操作异常	EXCPT_INTERNAL_ERROR

5.6.1.10.2 查询迭代器数据记录条数（getLength）

定义	
unsigned long getLength() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	查询迭代器中数据记录的总条数（注意这里指的是迭代器中数据记录的总条数，该值在迭代器生命周期内是不变的）
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR

5.6.1.10.3 销毁迭代器对象（destroy）

定义	
void destroy() raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	销毁迭代器对象
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR

5.6.2 通用接口模块（module common）

本接口是管理对象的通用操作接口，其他管理对象类接口模块都要从本接口派生

5.6.2.1 对象的任意能力（Capability_T）

定义	
typedef globaldefs::NameAndStringValue_T Capability_T;	
说明	
对象说明	对象的任意能力

5.6.2.2 对象的任意能力表述列表（CapabilityList_T）

定义	
typedef sequence<Capability_T> CapabilityList_T;	
说明	
对象说明	对象的任意能力表述列表

5.6.2.3 Common_I 接口

5.6.2.3.1 设置对象网管本地名（setNativeEMSName）

定义	
void setNativeEMSName(in globaldefs::NamingAttributes_T objectName, in string nativeEMSName) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	指定对象名称，设置该对象的本地名称
输入参数	objectName：对象名称。 nativeEMSName：分配给对象的新的 EMS 名称
输入/输出参数	无
输出参数	无
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.6.2.3.2 设置对象所有者（setOwner）

定义	
void setOwner(in globaldefs::NamingAttributes_T objectName, in string owner) raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	指定对象名称，设置该对象所有者名称
输入参数	objectName：对象名称。 owner：分配给对象的新的所有者名称
输入/输出参数	无
输出参数	无

说明	
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

5.6.2.3.3 设置对象用户标签（setUserLabel）

定义	
void setUserLabel (in globaldefs::NamingAttributes_T objectName, in string userLabel, in boolean enforceUniqueness) raises(globaldefs::ProcessingFailureException);	
说明	
功能描述	指定对象名称，设置该对象用户标签属性
输入参数	objectName：对象名称。 userLabel：分配给对象的新的用户标签。 enforceUniqueness：用户标签是否唯一
输入/输出参数	无
输出参数	无
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLO_COMPLY EXCPT_NE_COMM_LOSS

5.6.2.3.4 设置对象附加属性（setAdditionalInfo）

定义	
void setAdditionalInfo(in globaldefs::NamingAttributes_T objectName, inout globaldefs::NVSList_T additionalInfo) raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	设置对象附加信息
输入参数	objectName：对象名称
输入/输出参数	additionalInfo：对象附加信息
输出参数	无
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_NE_COMM_LOSS EXCPT_UNABLE_TO_COMPLY

5.6.2.3.5 查询对象能力描述（getCapabilities）

定义	
void getCapabilities(out CapabilityList_T capabilitiesList) raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	查询对象能力，例如是否支持交叉共享等(增加环回能力度)
输入参数	无
输入/输出参数	无
输出参数	capabilitiesList：对象的任意能力表述列表，类型为名值对
操作异常	EXCPT_INTERNAL_ERROR

5.6.3 通知管理模块（module notifications）

通知管理功能包括通知定购功能，通知上报功能和事件同步功能。本部分通知格式定义（心跳通知、告警通知除外）见TMF 814中的“OMGServicesUsage.pdf”；CosNotification::StructuredEvent的定义见“CosNotification.idl”。

心跳通知的事件头、事件体格式均应符合TMF 814中“OMGServicesUsage.pdf”的定义，其中可过滤体部分定义见下表。

名称	类型	取值描述
"notificationId"	string	通知标识符，用来在需要时唯一标识通知，可进行通知的关联
"object Name"	string	发送通知的对象名称
"objectType"	notifications::ObjectType_T	发送通知的对象类型
"emsTime"	globaldefs::Time_T	发送心跳通知的时间

告警通知的事件头、事件体格式均符合TMF 814中“OMGServicesUsage.pdf”的定义，其中可过滤体部分定义见下表。

名称	类型	取值描述
"notificationId"	string	通知标识符，用来在需要时唯一标识通知，可进行通知的关联
"objectName"	globaldefs::NamingAttributes_T	发送通知的对象名称
"nativeEMSName"	string	发送通知的对象的本地 EMS 名称
"nativeProbableCause"	string	本地可能的原因
"objectType"	notifications::ObjectType_T	发送通知的对象类型
"objectTypeQualifier"	notifications::ObjectTypeQualifier_T	对象类型指示符，标识对象是否为新创建的对象类型
"emsTime"	globaldefs::Time_T	告警开始的 EMS 时间
"neTime"	globaldefs::Time_T	告警开始的网元时间
"emsEndTime"	globaldefs::Time_T	告警结束的 EMS 时间，可选
"neEndTime"	globaldefs::Time_T	告警结束的网元时间，可选
"isClearable"	boolean	是否是可清除的事件
"layerRate"	transmissionParameters::LayerRate_T	层速率
"probableCause"	string	可能原因
"probableCauseQualifier"	string	告警原因定位标识（通过 objectName,layerRate,和 probableCause 唯一标识一个告警）

名称	类型	取值描述
"perceivedSeverity"	notifications::PerceivedSeverity_T	告警级别
"serviceAffecting"	notifications::ServiceAffecting_T	是否影响服务
"affectedTPList"	globaldefs::NamingAttributesList_T	影响的 TP 列表，可选
"additionalText"	string	附加文本
"additionalInfo"	globaldefs::NVSList_T	附加信息，可选
"X.733::EventType"	string	告警的基本类型，可选。共有如下 5 类："communicationsAlarm","environmentalAlarm","equipmentAlarm","processingErrorAlarm", "qualityofServiceAlarm"
"X.733::SpecificProblems"	notifications::SpecificProblemList_T	优化后的告警原因，可选
"X.733::BackedUpStatus"	string	备份状态，可选。取值为"BACKED_UP", "NOT_BACKED_UP"之一
"X.733::BackUpObject"	globaldefs::NamingAttributes_T	备份对象，提供通知备份服务的对象。该参数为条件必选，当"X.733::BackedUpStatus"取值为"BACKED_UP"，该参数有值
"X.733::TrendIndication"	string	变化趋势，可选。取值为"MORE_SEVERE", "NO_CHANGE", "LESS_SEVERE"当中的一个
"X.733::CorrelatedNotifications"	notifications::CorrelatedNotificationList_T	关联通知列表，可选
"X.733::MonitoredAttributes"	notifications::NVList_T	监视属性，可选
"X.733::ProposedRepairActions"	notifications::ProposedRepairActionList_T	故障处理的建议操作，可选
"X.733::AdditionalInfo"	notifications::NVList_T	X.733 附加信息，可选
"rcaiIndicator"	boolean	RACI 指示符，取值为"TRUE"表示该告警为根原因告警，取值为"FALSE"表示该告警为原始告警。缺省时默认取值为"FALSE"
"acknowledgeIndication"	notifications::AcknowledgeIndication_T	告警确认状态

5.6.3.1 告警级别（PerceivedSeverity_T）

定义

enum PerceivedSeverity_T {
PS_INDETERMINATE,
PS_CRITICAL,
PS_MAJOR,
PS_MINOR,
PS_WARNING,
PS_CLEARED
};

说明	
对象说明	告警级别
类型取值	取值说明
PS_INDETERMINATE	未确认
PS_CRITICAL	严重告警
PS_MAJOR	主要告警
PS_MINOR	次要告警
PS_WARNING	警告
PS_CLEARED	已清除

5.6.3.2 告警级别列表（PerceivedSeverityList_T）

定义	
typedef sequence<PerceivedSeverity_T> PerceivedSeverityList_T;	
说明	
对象说明	告警级别列表

5.6.3.3 告警标识（AlarmId_T）

定义	

struct AlarmId_T {
globaldefs::NamingAttributes_T objectName;
transmissionParameters::LayerRate_T layerRate;
ProbableCauseList_T probableCause;
string probableCauseQualifier;
};	

说明	
对象说明	告警标识
属性名	属性说明
objectName	对象名称
layerRate	速率层次
probableCause	告警可能原因
probableCauseQualifier	告警原因定位标识

5.6.3.4 告警/事件原因（ProbableCause_T）

定义	
typedef string ProbableCause_T;	
说明	
对象说明	告警/事件原因

5.6.3.5 告警/事件原因列表（ProbableCauseList_T）

定义	
typedef sequence < ProbableCause_T > ProbableCauseList_T;	
说明	
对象说明	告警/事件原因列表

5.6.3.6 对象的名称和取值（NameAndAnyValue_T）

定义	

struct NameAndAnyValue_T{
string name;
any value;
};	

说明	
对象说明	对象的名称和取值，这里的对象取值是 any 类型，本结构主要用于事件通知的通知结构中
属性名	属性说明
name	对象名称
value	对象取值

5.6.3.7 名值对列表（NVList_T）

定义
typedef sequence <NameAndAnyValue_T> NVList_T;
说明
对象说明	一个或多个对象的名称和取值对

5.6.3.8 事件信息列表（EventList_T）

定义
typedef sequence <CosNotification::StructuredEvent> EventList_T;
说明
对象说明	事件信息列表

5.6.3.9 对象类型（ObjectType_T）

定义

enum ObjectType_T {
OT_EMS,
OT_MANAGED_ELEMENT,
OT_MULTILAYER_SUBNETWORK,
OT_TOPOLOGICAL_LINK,
OT_SUBNETWORK_CONNECTION,
OT_PHYSICAL_TERMINATION_POINT,
OT_CONNECTION_TERMINATION_POINT,
OT_TERMINATION_POINT_POOL,
OT_EQUIPMENT_HOLDER,
OT_EQUIPMENT,
OT_PROTECTION_GROUP,
OT_TRAFFIC_DESCRIPTOR,
OT_AID
};

说明

对象说明	对象类型
类型取值	取值说明
OT_EMS	EMS
OT_MANAGED_ELEMENT	管理单元
OT_MULTILAYER_SUBNETWORK	子网
OT_TOPOLOGICAL_LINK	拓扑连接
OT_SUBNETWORK_CONNECTION	子网连接
OT_PHYSICAL_TERMINATION_POINT	物理终端点
OT_CONNECTION_TERMINATION_POINT	子网终端点
OT_TERMINATION_POINT_POOL	终端点池
OT_EQUIPMENT_HOLDER	设备容器
OT_EQUIPMENT	设备
OT_PROTECTION_GROUP	保护组
OT_TRAFFIC_DESCRIPTOR	通信描述符
OT_AID	未定义对象

5.6.3.10 对象类型指示符（ObjectTypeQualifier_T）

定义	
typedef string ObjectTypeQualifier_T;	
说明	
对象说明	对象类型指示符，标识对象是否为新创建的对象类型

5.6.3.11 告警确认状态（AcknowledgeIndication_T）

定义	

enum AcknowledgeIndication_T
{
AI_EVENT_ACKNOWLEDGED,
AI_EVENT_UNACKNOWLEDGED,
AI_NA
};	

说明	
对象说明	告警确认状态
类型取值	取值说明
AI_EVENT_ACKNOWLEDGED	确认
AI_EVENT_UNACKNOWLEDGED	未确认
AI_NA	未知

5.6.3.12 故障处理建议操作（ProposedRepairAction_T）

定义	
typedef string ProposedRepairAction_T;	
说明	
对象说明	故障处理建议操作

5.6.3.13 故障处理建议操作列表（ProposedRepairActionList_T）

定义	
typedef sequence< ProposedRepairAction_T > ProposedRepairActionList_T;	
说明	
对象说明	故障处理建议操作列表

5.6.3.14 告警原因优化（SpecificProblem_T）

定义	
typedef string SpecificProblem_T;	
说明	
对象说明	告警原因优化

5.6.3.15 告警原因优化列表（SpecificProblemList_T）

定义	
typedef sequence< SpecificProblem_T > SpecificProblemList_T;	
说明	
对象说明	告警原因优化列表

5.6.3.16 是否影响服务（ServiceAffecting_T）

定义
enum ServiceAffecting_T {

定义
SA_UNKNOWN, SA_SERVICE_AFFECTING, SA_NON_SERVICE_AFFECTING };
说明

对象说明	服务影响
类型取值	取值说明
SA_UNKNOWN	未知
SA_SERVICE_AFFECTING	影响服务
SA_NON_SERVICE_AFFECTING	不影响服务

5.6.3.17 通知 ID 的列表

定义
typedef sequence<string> NotifIDList_T;
说明

对象说明	通知 ID 的列表

5.6.3.18 关联通知（CorrelatedNotifications_T）

定义

struct CorrelatedNotifications_T
{
globaldefs::NamingAttributes_T source;
NotifIDList_T notifIDs;
};

说明

对象说明	关联通知
属性名	属性说明
source	发出关联通知的对象名称
notifIDs	关联通知的 ID

5.6.3.19 关联通知列表（CorrelatedNotificationList_T）

定义
typedef sequence<CorrelatedNotifications_T> CorrelatedNotificationList_T;
说明

对象说明	关联通知列表

5.6.3.20 EventIterator_I 接口

5.6.3.20.1 从迭代器中查询事件信息（next_n）

定义
boolean next_n (in unsigned long how_many, out EventList_T eventList) raises (globaldefs::ProcessingFailureException);

说明	
功能描述	从迭代器中查询 *n* 条记录。当 how_many≥迭代器剩余记录数时，系统在数据取完后应该自动销毁该迭代器对象
输入参数	how_many：首次迭代查询返回的数据数目
输入/输出参数	无
输出参数	eventList：首次查询返回的事件数据列表
操作异常	EXCPT_INTERNAL_ERROR

5.6.3.20.2 查询迭代器记录条数（getLength）

定义	
unsigned long getLength () raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	查询迭代器中数据记录的总条数（注意这里指的是迭代器中数据记录的总条数，该值在迭代器生命周期内是不变的）
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR

5.6.3.20.3 销毁迭代器对象（destroy）

定义	
void destroy () raises (globaldefs::ProcessingFailureException);	
说明	
功能描述	销毁迭代器对象
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR

附 录 A
（规范性附录）
模块与IDL的映射

本部分各模块与IDL文件间的映射关系如表A.1所示。

表A.1 模块与 IDL 的映射表

模块名称	IDL 名称	包含 IDL 名称
aSAP	aSAP.idl	globaldefs.idl notifications.idl
common	common.idl	globaldefs.idl
emsMgr	emsMgr.idl	common.idl multiLayerSubnetwork.idl notifications.idl aSAP.idl
emsSession	emsSession.idl	CosNotifyChannelAdmin.idl common.idl session.idl
emsSessionFactory	emsSessionFactory.idl	globaldefs.idl session.idl emsSession.idl nmsSession.idl mtnmVersion.idl
equipment	equipment.idl	common.idl globaldefs.idl terminationPoint.idl
flowDomain	flowDomain.idl	common.idl flowDomainFragment.idl topologicalLink.idl
flowDomainFragment	flowDomainFragment.idl	subnetworkConnection.idl Performance.idl
globaldefs	globaldefs.idl	--
maintenanceOps	maintenanceOps.idl	common.idl transmissionParameters.idl
managedElement	managedElement.idl	transmissionParameters.idl
managedElementManager	managedElementManager.idl	globaldefs.idl common.idl managedElement.idl transmissionParameters.idl terminationPoint.idl notifications.idl subnetworkConnection.idl aSAP.idl
mtnmVersion	mtnmVersion.idl	--

表 A.1（续）

模块名称	IDL 名称	包含 IDL 名称
multiLayerSubnetwork	multiLayerSubnetwork.idl	globaldefs.idl common.idl subnetworkConnection.idl transmissionParameters.idl managedElement.idl topologicalLink.idl terminationPoint.idl
nmsSession	nmsSession.idl	session.idl globaldefs.idl
notifications	notifications.idl	globaldefs.idl CosNotification.idl performance.idl transmissionParameters.idl
performance	performance.idl	transmissionParameters.idl common.idl
protection	protection.idl	globaldefs.idl transmissionParameters.idl common.idl
session	session.idl	
subnetworkConnection	subnetworkConnection.idl	globaldefs.idl transmissionParameters.idl terminationPoint.idl
terminationPoint	terminationPoint.idl	globaldefs.idl transmissionParameters.idl
timeMgr	timeMgr.idl	globaldefs.idl common.idl
topologicalLink	topologicalLink.idl	globaldefs.idl transmissionParameters.idl
trafficConditioningProfile	trafficConditioningProfile.idl	terminationPoint.idl globaldefs.idl common.idl transmissionParameters.idl transmissionDescriptor.idl
transmissionParameters	transmissionParameters.idl	globaldefs.idl

附 录 B
（规范性附录）
文件接口命名规则

B.1 文件的命名

文件的命名规则为：

原始的文件命名：<参考模型>-<日期与时间>[-P<ii>].xml或者<参考模型>-<日期与时间>[-P<ii>].csv

打包压缩后的文件命名：<参考模型>-<日期与时间>[-P<ii>].xml.[tar/zip].[gz]或者<参考模型>-<日期与时间>[-P<ii>].csv.[tar/zip].[gz]

文件名中各部分的取值说明如下：

1） <参考模型>：必选字段，标识配置文件遵循的网络资源模型标准。它可以进一步分解为[<网络资源模型发布方>-]<网元类型>-<网络资源模型类型>-<版本>

a） [<网络资源模型发布方>-]：可选字段，用于标识发布网络资源模型标准的单位，如“CCSA-”、“TMF-”等。当要同时启用不同来源的网络资源模型标准时，用此字段来区分文件。

b） <网元类型>：标识配置文件适用的网元类型，按照 EMS 所管理的网元类型来填写。如“OTN”、“PTN”、“WDM”、“SDH”，这四类任意两类及以上混合组网取值为“MIX”。

c） <网络资源模型类型>：取值为“NRM-<对象类型>”、“PM”或“FM”。对于配置文件仅可取 NRM-<对象类型>；对于性能文件仅可取 PM；对于告警文件仅可取 FM。

i. <对象类型>：取值如下：EMS、ME、EH、EQ、PTP、FTP、TL、SNC、SNCROUTE、FDFr、FDFrROUTE、PG、EPG、TNP、VRRP、FRR、StaticRouting、Bridge、VRF[注：FTP 信息包含在 PTP 对象类型的资源文件中]。

d） <版本>：标识文件遵循的信息模型的规范版本，允许在同一个目录下存放不同版本的信息模型规范的 NRM 文件。

2） <日期与时间>：必选字段，格式为“YYYYMMDD-HHMM”，指示数据的本地时间戳（注：此处不用指明时区）。

3） [P<ii>]：可选字段，当文件名指定的信息模型内容被分割存放到多个文件中时，第一部分为“P00”，第二部分为“P01”，以此类推。分割后的每个 XML 子文件必须符合本标准约定的文件格式要求。文件的分割可能是因为文件超过预设的大小（文件大小不约定），其原因不作限制。当出现文件分割时，NMS 需要采集所有子文件的数据才能保障数据完整性。

B.2 文件的打包

配置数据文件按照采集对象必须分别进行打包。性能文件和告警文件分别单独打包。打包后的文件名由原XML文件名后增加相应的打包文件后缀名构成。

B.3 文件的压缩

为了提升接口数据传输效率，缺省应将NRM文件压缩后传输。EMS应使用zip或gzip压缩格式对NRM文件进行压缩，压缩文件时不再分卷，压缩后的文件名由原XML文件名或者打包文件名之后增加相应的压缩文件后缀名构成。配置数据的打包文件必须分别进行压缩。

B.4 文件的更新

当由于某种原因导致原先生成的文件内容有误（如数据缺失等）时，EMS需要重新生成网络资源模型文件提供给NMS进行数据补采，新生成的文件在原文件名后增加”R<i>”，其中i每重新生成一次加一。

B.5 文件处理举例

文件处理举例见表 B.1。

表 B.1 文件处理举例

文件类型	原始文件名	打包后文件名	压缩后文件名
配置数据	CCSA-PTN-NRM-ME-V1.0.0-20140411-1602-P00.xml CCSA-PTN-NRM-ME-V1.0.0-20140411-1602-P01.xml	CCSA-PTN-NRM-ME-V1.0.0-20140411-1602.xml.tar/zip	CCSA-PTN-NRM-ME-V1.0.0-20140411-1602.xml.tar/zip.gz
历史告警	CCSA-PTN-FM-V1.0.0-20140411-1602-P00.xml CCSA-PTN-FM-V1.0.0-20140411-1602-P01.xml	CCSA-PTN-FM-V1.0.0-20140411-1602.xml.tar/zip	CCSA-PTN-FM-V1.0.0-20140411-1602.xml.tar/zip.gz
历史性能	CCSA-PTN-PM-V1.0.0-20140411-1602-P00.csv CCSA-PTN-PM-V1.0.0-20140411-1602-P01.csv	CCSA-PTN-PM-V1.0.0-20140411-1602.csv.tar/zip	CCSA-PTN-PM-V1.0.0-20140411-1602.csv.tar/zip.gz

中 华 人 民 共 和 国

通 信 行 业 标 准

分组传送网(PTN)网络管理技术要求

第 5 部分：基于 IDL/IIOP 技术的 EMS-NMS 接口信息模型

YD/T 2336.5—2016

*

人民邮电出版社出版发行

北京市丰台区成寿寺路 11 号邮电出版大厦

邮政编码：100164

北京康利胶印厂印刷

*

开本：880 × 1230　1/16　　2016 年 6 月第 1 版

印张：11　　2016 年 6 月北京第 1 次印刷

字数：306 千字

15115 • 1019

定价：110 元

本书如有印装质量问题，请与本社联系　电话：(010)81055492